МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Одеська політехніка»

ІНФОРМАТИКА ТА МАТЕМАТИЧНІ МЕТОДИ В МОДЕЛЮВАННІ

INFORMATICS AND MATHEMATICAL METHODS IN SIMULATION

Том 15, № 2

Volume 1, No. 2

Одеса – 2025
Odesa – 2025

Журнал внесений до переліку наукових фахових видань України (технічні науки) згідно наказу Міністерства освіти і науки України № 463 від 25.04.2013 р. Перереєстровано на категорію «Б» за фахами 121, 122, 125, 151 згідно наказу МОН України № 1473 від 26.11.2020 р.

Виходить 4 рази на рік

Заснований Одеським національним політехнічним університетом у 2011 році

Свідоцтво про державну реєстрацію КВ № 17610 - 6460Р від 04.04.2011р.

Published 4 times a year

Founded by Odesa National Polytechnic University in 2011

Certificate of State Registration КВ № 17610 - 6460Р of 04.04.2011

Друкується за рішенням редакційної колегії та Вченої ради Національного університету «Одеська політехніка»

Оригінал-макет виготовлено редакцією журналу

Адреса редакції: 1, Шевченка пр., Одеса, 65044, Україна
Телефон: +38 048 705 8506
Web: www.immm.op.edu.ua (immm.opu.ua) E-mail: immm.ukraine@gmail.com

Editorial address: 1, Shevchenko Ave., Odesa, 65044, Ukraine
Tel.: +38 048 705 8506
Web: www.immm.op.edu.ua (immm.opu.ua) E-mail: immm.ukraine@gmail.com

ЗМІСТ/CONTENTS

DOI 10.15276/imms.v15.no2.155
UDC 004.9

Informatics and Mathematical Methods in Simulation
Vol.15 (2025), No. 2, pp. 155-165

INTEGRATED APPROACH TO CREATING A CASE-BASED DATABASE FOR DIAGNOSING FAILURES IN SHIP POWER PLANTS

V.V. Vychuzhanin, A.V. Vychuzhanin

National Odesa Polytechnic University
1, Shevchenko Ave., Odesa, 65044, Ukraine
Email: v.v.vychuzhanin@op.edu.ua

Modern ship power plants (SPPs) represent highly complex technical systems operating under variable, high-load, and often harsh maritime conditions. These systems are characterized by a high degree of component interdependence, making the identification and prediction of failures a challenging task. Traditional deterministic diagnostic approaches often fail to accurately reflect the stochastic nature of technical failures and cascading effects. This article presents a comprehensive integrated approach to failure diagnostics based on a case-based reasoning (CBR) system. The core of the methodology is a structured case base that unites historical failure data, probabilistic models (including Bayesian networks and Markov processes), and discrete-event simulation tools. The structure of each case is standardized and includes failure descriptions, associated risk levels, diagnostic method references, operational context, and interconnection effects. The database currently includes over 5,000 unique failure scenarios derived from operational logs, maintenance records, and simulated data. To increase diagnostic accuracy and system adaptability, the authors implement dynamic updates and automated optimization of parameter weights using the L-BFGS-B algorithm. Simulation modeling is applied to reproduce rare and cascading failure conditions and to improve the generalization capacity of the system. Numerical experiments demonstrate that this integrated approach achieves a diagnostic accuracy of up to 95%, reduces false positives by 30%, and increases system flexibility in real-time operational contexts. The fusion of CBR with probabilistic and simulation models enables the system not only to diagnose known patterns but also to predict new and atypical failures, accounting for system degradation over time. The result is a knowledge-driven support tool for decision-making in ship power plant operations, significantly enhancing operational safety and maintenance planning. This work has implications for the development of intelligent diagnostic platforms in complex marine and industrial systems.

Keywords: fault diagnostics; failure prediction; stochastic processes; intelligent systems; operational risks; data analytics

Introduction. Modern ship power plants (SPPs) are complex technical systems (CTS) operating under harsh operational conditions [1]. The reliability and efficiency of SPP operation largely depend on the timely diagnosis and prediction of equipment failures [2]. However, traditional diagnostic methods based on deterministic models or expert assessments often lack flexibility and fail to account for the complexity of multicomponent systems [3, 4].

One of the promising approaches for diagnosing failures in CTS equipment is Case-Based Reasoning (CBR), which enables the use of accumulated experience to identify similar failures and predict their possible consequences [5, 6]. However, existing CBR systems face several challenges, such as optimizing the case database, accounting for probabilistic dependencies between parameters, and reducing computational load when processing large volumes of data.

An analysis of existing CBR optimization approaches shows that significant efforts have been directed toward improving case retrieval methods and database management. For instance, the study [7] proposes an enhanced case retrieval method in CBR systems for diagnosing aircraft engines. This approach considers attribute interactions, improving the accuracy of equipment condition diagnostics. However, its high computational complexity and limited applicability (aviation engines) make it less universal. The study [8] introduces a

dynamic case base maintenance method that optimizes database size without loss of accuracy. However, this method does not address case interaction effects or performance efficiency with large data volumes. Researchers in [9] presented a case base and feature dictionary update method based on the theory of belief functions. This method's advantage is the automatic adjustment of the knowledge base size, though its implementation complexity and potential increase in computational load remain open issues. Ayed et al. developed a method for removing duplicate cases in the case base, but it requires manual tuning [10].

Thus, despite extensive research in CBR optimization, the integration of dynamic updating and uncertainty processing remains a relevant challenge. For effective CBR application, a structured case base must be developed that includes not only historical failure data but also probabilistic estimates, information on cascading effects, operational parameters, and simulation-based failure modeling results for CTS.

The aim of this study is to develop and justify an integrated approach to diagnosing failures in marine power plants by creating a case-based reasoning (CBR) database that combines probabilistic models and simulation modeling. This approach enables consideration of the stochastic nature of failures, cascading effects, and dynamic changes in operational conditions, thereby improving the accuracy of diagnostics and failure prediction.

Main part. Case Structure. To standardize the representation of failures, a case structure has been developed, including: failure description (type, causes, consequences); failure risk assessment (Harrington's desirability function [11], probabilistic failure assessment); component characteristics (unit condition, remaining resource, failure intensity); diagnostic methods (CBR, Bayesian networks [12, 13, 14], simulation modeling [15]); data sources (maintenance logs, OREDA databases [16], expert assessments).

As an example, the structure of a single case can be presented in Table 1.

Table 1.

Case structure of SPP equipment failures

Parameter	Description	Method/Source
Failure type	Failure code(1 — mechanical *2* — electrical etc.)	Expert assessment, technical documentation
Failure causes	List of identified failure causes	Log analysis, expert opinion
Failure Consequences	Description of the impact on the system (efficiency reduction, accident risk, etc.)	Technical documentation, reports
Failure risk (desirability)	Value based on desirability function (0-1)	Harrington function calculation
Failure probability	Failure probability assessment (e.g. 0.05))	Statistical data (OREDA)
Unit Condition (Si)	0 (operational). 1 (degradation). *2* (pre-failure). 3 (failure)	Expert assessment condition sensors
Remaining resource (Ri(t))	Assessment of the remaining service life of the unit	Markov models
Diagnostic methods	Applied algorithms and similarity measures	CBR. Bayesian networks, simulation modeling

The formalized structure of a failure precedent can be represented as a JSON object. This format is well suited for machine processing and visual representation of relationships:

```
{
 "failure_case": {
  "identifier": {
   "code": "UNQ-12345",
   "date": "2025-03-19",
   "source": "OREDA"
  },
  "failure_type": {
   "category": "mechanical",
   "causes": ["wear", "fatigue damage"]
  },
  "operational_context": {
   "working_conditions": {
    "temperature": "75°C",
    "vibration": "increased",
    "load": "90%"
   },
   "operating_mode": "overload",
   "external_factors": ["oil contamination", "low fuel quality"]
  },
  "failure_risk": {
   "probability": 0.15,
   "risk_category": {
    "method": "Harrington",
    "level": "high"
   },
   "expected_damage": {
    "cost": 15000,
    "safety_impact": "critical"
   }
  },
  "interconnected_components": {
   "subsystems": ["fuel system", "hydraulics"],
   "connection_type": {
    "mechanical": true,
    "electrical": false,
    "informational": true
   },
   "cascade_effects": ["pump damage", "overheating"]
  },
  "data_sources": {
   "historical": ["OREDA", "maintenance logs"],
   "simulation": ["Bayesian networks", "discrete-event modeling"],
   "sensor_based": {
    "IoT": true,
    "SCADA": true
   },
   "expert_assessments": ["maintenance engineers", "diagnostic reports"]
  },
  "diagnostic_methods": {
   "CBR": {
    "case_retrieval": "k-NN",
    "adaptation": "gradient methods"
   },
   "Bayesian_networks": {
    "analysis": "probabilistic"
   },
```

```
    "hybrid_methods": {
      "CBR_and_machine_learning": ["L-BFGS-B", "regression"],
      "CBR_and_simulation": ["Markov processes", "Bayesian networks"],
      "weight_optimization": ["gradient methods"]
    }
   }
  }
 }
}
```


The presented code reveals: structuredness – data connections are interrelated; flexibility – easily expandable with new parameters; processing readiness – applicable in diagnostic and analysis systems.

This structure includes key failure parameters, data sources, and diagnostic methods, enabling the standardization of the case database and improving the efficiency of searching for similar cases.

Case Database Formation.

Updating the knowledge base includes recalculating failure probabilities, filtering data, and dynamically adapting the diagnostic system. The main data categories are presented in Table 2.

Table 2.

Main Data Categories in the Knowledge Base

Data Category	Description	Data Source
Case Database	Historical data on failures, their causes, and consequences	Operational data, maintenance logs, operational reports
Probabilistic Failure Models	Estimates of failure probabilities and their cascading effects	Failure statistics, Bayesian networks. expert assessments
Simulation Scenarios	Simulated failure situations, including rare and cascading events	Discrete-event modeling, Monte Carlo method, cognitive models
Repair and Maintenance Data	Information on performed repairs and technical maintenance	Technical documentation, operational logs
Anomalies and Verification	Data on identified deviations in equipment operation	Analysis of real operational data, statistical anomaly control
Adaptive CBR Parameters	Adjustment of parameter weights during analogy search to improve diagnostic accuracy	Dynamic training on new data

The developed case database includes over 5,000 records of ship power plant equipment failures collected from operational reports, technical inspections, and emergency situations. The database structure provides information on failure types, operating conditions, probabilistic characteristics, and diagnostic procedures.

The most common failure categories include fuel system malfunctions (27% of cases), bearing failures (18%), gas turbine overheating (15%), and anomalies in electrical system operation (12%).

The automatic knowledge base update process consists of several key stages: adjusting failure probabilities, including recalculations based on Bayesian analysis; filtering data to eliminate unreliable information; and adapting diagnostic models, including parameter weight adjustments in CBR.

Simulation Model for Predicting the Technical Condition of Ship Power Plant Equipment.

To improve the accuracy of predicting the technical condition of SPP equipment, a simulation model is used, which includes: modeling of component failures (Weibull and exponential distributions); consideration of cascading failure effects; analysis of operational modes and maintenance procedures.

The relationships between the model components are shown in Figure 1. The comprehensive SPP schematic (Fig. 1) [17] includes not only the main power units (e.g., the main engine, power plant) but also auxiliary systems that ensure their operation and the safety of the vessel.

The following relationships can be established between the components and subsystems in Figure 1: main engine corresponds to the main engine (8); cooling system is not explicitly specified in the structure but may be part of the remote-automated control subsystem (9) or included in the main engine support system; fuel system is not directly highlighted but is logically linked to the boiler plant (5) and the main engine (8); generator may be part of the ship's power plant (6); pumping system is likely connected to the ballast and bilge subsystem (10) and other SPP support systems; power supply corresponds to the ship's power plant (6); automation system may relate to the remote-automated control subsystem (9).

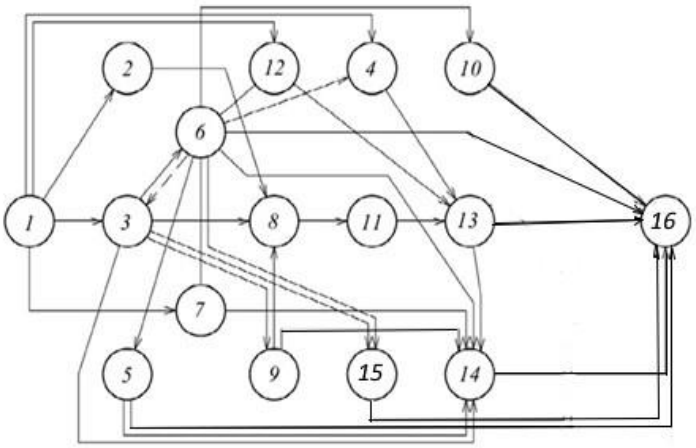


Fig.1. Structure of the SPP: input component 1; manual control of the main engine 2; compressed air subsystem 3; propulsion and steering control subsystem (PSC) 4; boiler plant 5; ship power plant 6; fire protection system 7; main engine 8; remote-automated control subsystem 9; ballast and bilge subsystem 10; power transmission from the main engine to the propeller 11; emergency PSC drive 12; PSC 13; measuring instruments subsystem 14; sanitary water treatment subsystem 15; output component 16.

The key parameters of the simulation scenarios for SPP equipment operation are presented in Table 3. Table 3 contains the parameters of failure simulation scenarios, allowing for an analysis of system behavior under various operating conditions.

Table 3.

Parameters of Simulation Scenarios for SPP Equipment Operation

Scenario	Failure Rate (avg.)	Load Level	Possible Consequences
Normal Conditions	Low (0.01 failures/h)	Normal (70%)	Insignificant efficiency reduction
Accelerated Wear	Medium (0.05 failures/h)	High (90%)	Accelerated wear of key components
Critical Failures	High (0.1 failures/h)	Extreme (100%)	Cascading failures, SEU failure

The analysis of accumulated data has made it possible to establish the dependence of the probability of SEU equipment failures on operating time, as shown in Figure 2.

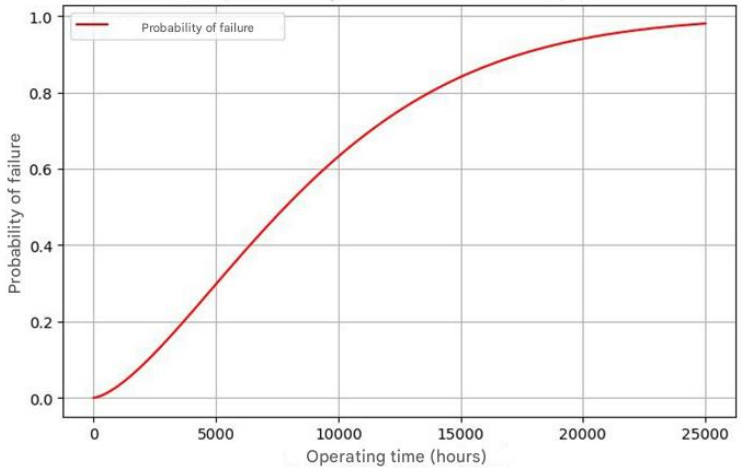


Fig. 2. Probability of Failure with Increasing Operating Time of SPP Equipment

The graph illustrates the increase in failure probability with extended equipment operation. In the initial phase (up to 5000 hours), the probability of failure is low, corresponding to the period of normal operation. After 15,000 hours, the probability of failure rises more sharply, indicating the need for more thorough maintenance. By 25,000 hours, the probability approaches 1, meaning the equipment is almost certain to fail. The curve demonstrates the characteristic growth in failure probability with increased operating time, confirming the necessity of using probabilistic models and simulation modeling in developing the case base. The obtained data confirm that accounting for the stochastic nature of failures and integrating the CBR method with probabilistic models not only enables failure prediction but also enhances diagnostic accuracy at various stages of the equipment life cycle.

Integration of Diagnostic Methods

The simulation model incorporates Bayesian networks for analyzing failure interdependencies, Markov processes [18] for modeling state transitions of the equipment, and cognitive modeling that accounts for expert knowledge and operational conditions. To improve diagnostic accuracy, CBR is used to identify similar failures, Bayesian networks to assess probabilistic dependencies, Markov processes to predict degradation, and simulation modeling to generate rare failures.

Table 4 presents the distribution of diagnostic methods and their functional purposes.

Table 4.

Distribution of Diagnostic Methods and Their Functional Purposes

Method	Analyzed Elements	Expected Result
CBR	Components similar to recorded failures	Finding precedents with similar symptoms
Bayesian Networks	Interconnected system elements	Probabilistic assessment of cascading failure effects
Markov Models	Remaining resource of components	Prediction of component degradation
Simulation Modeling	Failure dynamics in the system	Generation of artificial precedents to supplement the database

Figure 3 presents a comparison of equipment failure probabilities when using traditional diagnostic methods versus employing a case-based database integrated with the CBR method. It is evident that utilizing the case-based database reduces the probability of diagnostic errors, as accumulated failure experience is used to refine predictions. This demonstrates that the proposed approach not only enhances diagnostic accuracy but also improves the system's adaptation to changing operating conditions.

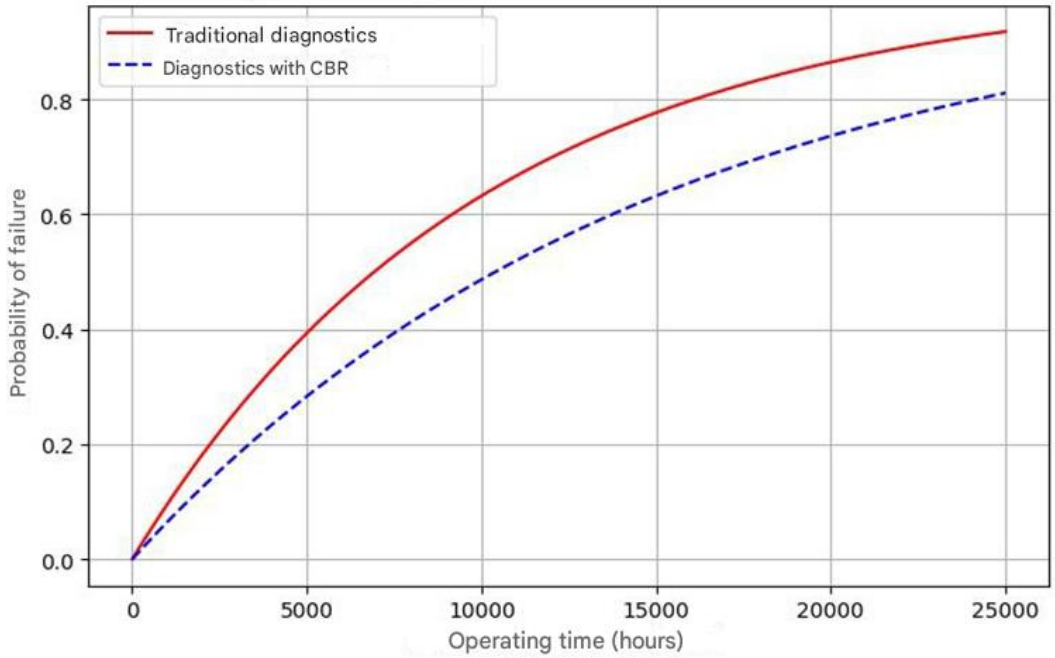


Fig. 3. Probability of Equipment Failures in SPP with and without CBR

The graphs illustrate the effect of applying CBR, which manifests in a reduction in diagnostic errors and an increase in prediction accuracy. The connection to the case-based database reflects the stochastic nature of failures: the graphs show how the probability of equipment failure increases over time. This confirms the importance of considering stochastic factors in diagnostics, which is achieved through the integration of probabilistic models in the case-based database.

The advantage of CBR over traditional diagnostic and failure prediction methods for SEU equipment is evident in the differences in failure forecasts: traditional approaches do not utilize accumulated experience, leading to a higher probability of diagnostic errors. In contrast, CBR analyzes historical data and adapts diagnostic decisions, reducing the likelihood of errors. This confirms that the case-based database refines predictions by incorporating previously recorded failures.

Simulation modeling enhances forecasting accuracy by accounting for not only frequent but also rare, cascading failures. Utilizing accumulated data improves the prediction of potential malfunctions, reducing the number of false alarms and increasing diagnostic accuracy.

The integration of these methods ensures not only precise fault diagnostics but also system adaptability to new operating conditions, increasing prediction reliability and minimizing the likelihood of missed failures.

Numerical experiments have shown that applying the case-based database in combination with CBR adaptation increases diagnostic accuracy from 85% (when using only traditional probabilistic methods) to 92–95%. The integration of Bayesian networks and simulation modeling reduced the number of false-positive diagnostic decisions by 30%, while

using optimization algorithms for parameter weighting reduced the mean absolute error in failure prediction by 12% (Table 5).

Table 5.

Impact of the Case-Based Database on Failure Diagnosis Accuracy

Diagnostic Method	Accuracy %	FP (False Positives)	FN (False Negatives)	Processing Time **(s)**
Probabilistic Methods (without CBR**)**	85	18%	12%	2.1
CBR	89	14%	10%	18
Integrated Approach	95	8%	5%	1,2

Optimization of the Case Base

To enhance diagnostic accuracy and system adaptability, the case base is regularly updated and optimized. One of the key mechanisms is the adjustment of parameter weights that determine the degree of similarity between failure cases. Optimization is performed using the L-BFGS-B method, which minimizes the error between reference and predicted failure similarity values:

$$\omega^* = \arg min_{\omega} \sum_{i} (S_{true}(i) - S_{pred}(i, \omega))^2$$

where $S_{true}(i)$ is the reference similarity value between cases (based on expert assessments or failure statistics);

$S_{pred}(i, \omega)$ is the system-predicted similarity value, dependent on parameter weights ω, ω represents the diagnostic parameter weights, defining the contribution of various factors to f ailure similarity assessment

The L-BFGS-B (Limited-memory Broyden-Fletcher-Goldfarb-Shanno with Box constraints) method is a modification of the quasi-Newton optimization method BFGS that: allows minimization of a function dependent on a large number of parameters; uses an approximate inverse Hessian matrix representation to save memory; supports constraints on parameters, which is essential when optimizing weights (e.g., ensuring weights remain positive and sum to 1).

The main steps in updating the case base include: dynamic adjustment of parameter weights based on new operational data; filtering out outdated and irrelevant data to maintain database relevance; adapting the case base structure considering the results of simulation modeling.

Applying this approach ensures more accurate matching of failure cases and improves the diagnostic system's adaptability to changing operating conditions.

Conclusions. The presented approach to case base development justifies the necessity of structured failure data storage and the creation of a unified knowledge base that enables efficient failure diagnostics and prediction. Within the study, a precedent structure has been developed, incorporating key failure parameters, their probabilistic assessments, and links to operational factors. A knowledge base concept has been formulated, integrating the case base, probabilistic models, and failure simulation scenarios. Methods for automatic data updates have been defined, including CBR adaptation, failure probability recalculations, and data verification, ensuring system relevance. The integration of probabilistic and simulation

methods allows consideration of cascading failure effects and the prediction of rare malfunctions, significantly improving diagnostic accuracy.

The developed case base integrates the CBR method with probabilistic models and simulation modeling, enhancing the accuracy of ship power plant failure diagnostics. The analysis of the proposed approach has shown that using the case base increased diagnostic accuracy to 95% due to the accumulation and structuring of failure experience, while probabilistic methods ensure the consideration of the stochastic nature of failures, which is particularly critical for complex technical systems. Simulation modeling enabled the consideration of rare and cascading failures, expanding prediction capabilities. The developed case base covers the main types of failures in ship power plants, including fuel system malfunctions, gas turbine engine overheating, and electrical system anomalies. The process of accumulating and analyzing cases contributes to identifying failure patterns and their early prediction, reducing the number of false positives by 30%.

Thus, the developed case base serves as a key element of an intelligent ship power plant failure diagnostics system. It facilitates diagnostic data accumulation, decision support, enhanced failure prediction accuracy, as well as consideration of the stochastic nature of failures and cascading effects. The integrated approach, combining CBR, probabilistic models, and simulation modeling, significantly improves diagnostic efficiency and the reliability of ship power plants.

References

1. Vychuzhanin V., Vychuzhanin A. Stochastic Models and Methods for Diagnostics, Assessment, and Prediction of the Technical Condition of Complex Critical Systems: monograph. Lviv-Torun: Liha-Pres, 2025. 176 c. DOI: 10.36059/978-966-397-457-6.
2. Zhang P., GaoCao Z., Dong L. Marine Systems and Equipment Prognostics and Health Management. Systematic Review from Health Condition Monitoring to Maintenance Strategy. Machines. 2022. № 10. C. 72. DOI: 10.3390/machines10020072.
3. Vychuzhanin V., Rudnichenko N. Assessment of risks structurally and functionally complex technical systems. Eastern-European Journal of Enterprise Technologies. 2014. № 1(2). C. 18–22. DOI: 10.15587/1729-4061.2014.19846.
4. Vychuzhanin V., Rudnichenko N. Complex Technical System Condition Diagnostics and Prediction Computerization. CMIS-2020 Computer Modeling and Intelligent Systems. Ceur-ws.org. 2020. № 2608. C. 1–15. DOI: 10.32782/cmis/2608-4.
5. Glukhikh D., Shchinnikov I., Glukhikh I. Using hybrid-CBR for intelligence monitoring and decision-making systems on SMART grid. Intelligent Decision Technologies. 2022. № 16. C. 1–8. DOI: 10.3233/IDT-210239.
6. Aamodt A., Plaza E. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. AI Communications. IOS Press. 1994. T. 7, № 1. C. 39–59. DOI: 10.3233/AIC-1994-7104.
7. Chen M., Xia J., Huang R., Fang W. Case-Based Reasoning System for Aeroengine Fault Diagnosis Enhanced with Attitudinal Choquet Integral. Applied Sciences. 2022. № 12. C. 5696. DOI: 10.3390/app12115696.
8. Salamó M., Golobardes E. Dynamic Case Base Maintenance for a Case-Based Reasoning System. Lecture Notes in Computer Science. 2004. DOI: 10.1007/978-3-540-30498-2_10.
9. Ben Ayed S., Elouedi Z., Lefevre E. An evidential integrated method for maintaining case base and vocabulary containers within CBR systems. Information Sciences. 2020. T. 529. C. 214–229. DOI: 10.1016/j.ins.2019.11.009.

10. Ayed S.B., Elouedi Z., Lefevre E. CIMMEP: constrained integrated method for CBR maintenance based on evidential policies. Applied Intelligence. 2022. DOI: 10.1007/s10489-020-02159-4.
11. Harrington E.C. The desirability Function. Industrial Quality Control. 1965. № 21(10). C. 494–498.
12. Fam M.L., He X., Konovessis D. Dynamic Bayesian Belief Network for long-term monitoring and system barrier failure analysis: Decommissioned wells. MethodsX. 2022. № 9. C. 1–10. DOI: 10.1016/j.mex.2021.101600.
13. Jensen F.V. Bayesian networks basics. Tech. Rep. Denmark: Aalborg Univ., 1996. 12 c.
14. Wang Chen R., Guan C. A Bayesian inference-based approach for performance prognostics towards uncertainty quantification and its applications on the marine diesel engine. ISA Transactions. 2021. № 118. C. 159–173.
15. Vychuzhanin V., Rudnichenko N. Cognitive Model of the Internal Combustion Engine. SAE Technical Paper. 2018. № 2018-01-1738. 10 c. DOI: 10.4271/2018-01-1738.
16. OREDA-Offshore Reliability Data Handbook. 6th Edition. OREDA, 2015. 831 c.
17. Pan P., Sun Y., Yuan C., Yan X., Tang X. Research progress on ship power systems integrated with new energy sources: A review. Renewable and Sustainable Energy Reviews. 2021. № 144. DOI: 10.1016/j.rser.2021.111048.
18. Lisnianski A., Elmakias D., Laredo D., BennHaim H. A multistate Markov model for a short-term reliability analysis of a power generating unit. Reliability Engineering & System Safety. 2012. № 98. C. 1–6. DOI: 10.1016/j.ress.2011.10.008.

ІНТЕГРОВАНИЙ ПІДХІД ДО СТВОРЕННЯ БАЗИ ПРЕЦЕДЕНТІВ ДЛЯ ДІАГНОСТИКИ ВІДМОВ СУДНОВИХ ЕНЕРГЕТИЧНИХ УСТАНОВОК

В.В. Вичужанін, А.В. Вичужанін

Національний університет «Одеська політехніка»
1, Шевченка пр., м.Одеса, 65044, Україна
Email: v.v.vychuzhanin@op.edu.ua

Сучасні суднові енергетичні установки (СЕУ) є надзвичайно складними технічними системами, що функціонують в умовах високих навантажень, змінних середовищних параметрів та значної взаємозалежності між підсистемами. Ці фактори значно ускладнюють процеси своєчасної діагностики та прогнозування відмов. Традиційні детерміновані методи виявляються недостатньо ефективними, оскільки не враховують стохастичний характер збоїв і каскадну природу наслідків. У статті запропоновано інтегрований підхід до створення системи діагностики на основі методу прецедентів (CBR). У його основі лежить уніфікована база прецедентів, що включає історичні дані, ймовірнісні моделі (баєсівські мережі, марковські процеси) та результати імітаційного моделювання. Структура кожного прецеденту передбачає опис типу відмови, оцінку ризиків, умови експлуатації, джерела даних, методи діагностики та взаємозв'язки між компонентами. Сформована база містить понад 5000 випадків відмов, отриманих із експлуатаційної документації, журналів обслуговування та результатів моделювання. Для підвищення точності діагностики та адаптивності системи застосовано механізм автоматичного оновлення, оптимізації вагових коефіцієнтів (метод L-BFGS-B), а також постійне доповнення бази новими сценаріями збоїв. Імітаційне моделювання дозволяє враховувати як часті, так і рідкісні або каскадні відмови, формуючи повну картину можливих технічних ризиків. Результати чисельних експериментів підтверджують ефективність підходу: точність діагностики досягає 95%, зменшується частка хибнопозитивних результатів на 30%, а сама система виявляє гнучкість до змін умов експлуатації. Інтеграція CBR з ймовірнісними та імітаційними методами дозволяє прогнозувати розвиток відмов на різних етапах життєвого циклу обладнання, знижуючи ризики експлуатації та оптимізуючи планування обслуговування. Запропонована база знань є ефективним інструментом підтримки прийняття рішень у сфері морської енергетики й може бути адаптована для інших технічно складних систем.

Ключові слова: діагностика несправностей, прогнозування відмов, стохастичні процеси, інтелектуальні системи, експлуатаційні ризики, аналітика даних.

DOI 10.15276/imms.v15.no2.166
UDK 62-501,4:681.326.7

Informatics and Mathematical Methods in Simulation
Vol.15 (2025), No. 2, pp. 166-175

PLANNING OF THE DIAGNOSTIC EXPERIMENT IN THE LOCALIZATION OF TROUBLESHOOTING FAILURES OF SINGLE-FREE SYSTEMS

A.A. Savelev, L.L. Prokofieva

National Odesa Polytechnic University
1, Shevchenko Ave., Odesa, 65044, Ukraine
Emails: saveleva@op.edu.ua, prokofieva1957@gmail.com

Formalized conditions for carrying out a diagnostic experiment associated with the identification of faulty fragments (subschemes) of inertial-free systems are obtained. The diagnostic experiment is reduced to computational procedures for localization of faulty subschemes, which are based on testing hypotheses that the characteristics of the allocated subschemes have changed. Hypotheses are formulated in such a way as to ensure the detection of parametric and structural faults. The first, for example, may include changing the resistance of the circuit, and the second - an open or short circuit.
When the system is divided into subschemes, the latter are selected from the condition of their parametric identifiably, i.e. The situation when, based on the known parameters of the remaining subschemes, as well as the input and output signals in the whole system, it is possible to determine the parameters of the subscheme under consideration. Planning a diagnostic experiment is as follows. Assuming a linear relationship between the parameters of the allocated subscheme S_i and the output signals of the system, a verification matrix $\mathbf{F}$ is created in advance, which determines the interrelation of the indicated parameters and signals, considering the serviceability of the subscheme under consideration. In the event of a malfunction, the verification matrix $\mathbf{F}$ changes, which allows to determine the nonconformity matrix $\mathbf{q}$, on the basis of which analysis it is possible to obtain an estimate $\Delta\hat{\rho}_i$ of the parameters $\Delta\rho_i$ of the subscheme S_i under consideration, as a result of which it is concluded that it is serviceable. Procedures for testing hypotheses on the health of the subschemes of the inertial-free system for cases of parametric and structural faults are of an identical nature. The principal difference between experiment planning and structural faults from planning for parametric faults is that for parametric faults the value $\Delta\rho_i$ does not depend on the input signals of the system, while for structural faults — the value $\Delta\rho_i$ depends on the input signals of the system, in an unknown way.
Keywords: diagnostics, diagnostic experiment, fault location, inertial-free systems, estimation of subschemes parameters.

Introduction. To solve a wide range of problems in the study of systems (identification, control of operability, preventive maintenance), methods of the theory of experimental design are widely used, allowing, in the presence of measurement noise, to obtain estimates of the parameters of the system model that are best in some sense. It should also be taken into account that the features of the dynamic behavior of systems, for example, their lack of inertia, do not have a significant effect on the procedures for carrying out these experiments [1].

When diagnosing, even in the case of the most simple parametric faults in the subschemes, it is necessary first of all not to get the parameter estimates, but to establish the faulty subschemes. This formulation of the problem presupposes the need to ensure the best distinguish ability of the subschemes [2 — 4].

Objective. Formalization of the conditions for planning a diagnostic experiment in problems of determining the operability of inertial-free systems in the localization of parametric and structural faults in their subschemes.

Main part. Suppose that in one of the subschemes only parametric faults are possible. In this case, the description of the faulty subscheme S_i has the following form

$$Z_i = \varphi_i^*(V_i, \boldsymbol{\rho}_i);\ i = \overline{1, N},$$

where Z_i — is the vector of the subschemes S_i parameters, $\boldsymbol{\rho}_i = \boldsymbol{\rho}_i^* + \Delta\boldsymbol{\rho}_i$; $\boldsymbol{\rho}_i \in D_p$, D_p — the area of the values of the subscheme S_i parameters, $\boldsymbol{\rho}^*$ — the vector of the nominal values of the dimension r_i parameters, N — the number of subschemes to be checked.

A subscheme S_i is *parametrically identifiable* if, with known parameters of the remaining subschemes, the input and output signals in the system as a whole can be determined by its subscheme S_i parameters.

Consider the subscheme S_i for which the relationship between the parameters and the output signals of the system is linear. In this case, the description of a system with a defective subscheme S_i can be represented as follows:

$$\mathbf{y} = \mathbf{F}^*(\mathbf{u}) + \frac{\partial \mathbf{F}^*(\mathbf{u})}{\partial \boldsymbol{\rho}_i^{\mathrm{T}}} \Delta\boldsymbol{\rho}_i,\ i = \overline{1, N}, \tag{1}$$

where $\mathbf{y}$ — the vector of the output signals of the system, $\mathbf{u}$ — the vector of the input signals of the system, $\mathbf{F}^*(\mathbf{u})$ — a verification matrix composed of the elements of the fault dictionary.

Denoting $\left\{\left[\partial \mathbf{F}^*(\mathbf{u})\right] / \partial \boldsymbol{\rho}_i^{\mathrm{T}}\right\} = \mathbf{q}_i(\mathbf{u})$, we rewrite (1) in the form

$$\Delta y = \mathbf{y} - F^*(\mathbf{u}) = \mathbf{q}_i(\mathbf{u})\Delta\rho_i,\ i = \overline{1, N}. \tag{2}$$

If there is such a value $\mathbf{u} \in D_u$ as rank $q_i(\mathbf{u}) = r_i$, then from (2) we can obtain an estimate $\Delta\hat{\boldsymbol{\rho}}_i$ of the parameters $\Delta\boldsymbol{\rho}_i$. In this case, the identification task is solved with one set of input signals of the system. If the rank $q_i(\mathbf{u}) < r_i$, then the task of identification can be solved only if on the diagnosed system it is possible to give sets of input signals $\mathbf{U}^\kappa = \left(u^1, \ldots, u^\kappa\right)^{\mathrm{T}}$, in which

$$\operatorname{rank} \mathbf{Q}_i(U^\kappa) = \operatorname{rank} \begin{bmatrix} q_i(u^1) \\ \vdots \\ q_i(u^\kappa) \end{bmatrix} = r_i.$$

Estimation of the parameters of the subsysteme S_i is determined from equation

$$\Delta Y\left(U^\kappa\right) = \mathbf{Q}_i\left(U^\kappa\right)\Delta\rho_i;\ i = \overline{1, N}, \tag{3}$$

where $\Delta Y\left(U^\kappa\right) = \left[\Delta y\left(u^1\right), \ldots, \Delta y\left(u^\kappa\right)\right]^{\mathrm{T}}$.

The problem of diagnosing will be solved by testing hypotheses H_{ρ_i}. The hypothesis H_{ρ_i}, $i = \overline{1, N}$ is the assumption that the parameters of the subscheme S_i have changed and the description of the system has the form (1).

The hypothesis H_{ρ_i} is verified by testing the compatibility of equation (3), whose matrix $\mathbf{Q}\left(U^\kappa\right)$ must be rectangular.

Subschemes S_i, S_j; $i \neq j$ are not distinguishable under the hypothesis H_{ρ_i}, if for a given value $\Delta\rho_j$ there exists such a value $\Delta\rho_i \in D_p$ that

$$q_i(\mathbf{u})\Delta\rho_i = q_j(\mathbf{u})\Delta\rho_j,\ \mathbf{u} \in D_u. \tag{4}$$

The criteria for the distinguish ability of the subschemes S_i, S_j parameters are determined from the conditions under which equality (4) is not satisfied.

The parameters of the subschemes S_i, S_j are distinguishable under the hypothesis H_{ρ_i} if and only if there exists a value U^κ such that

$$\operatorname{rank} \left[\begin{array}{c|c} \mathbf{Q}_i(U^\kappa) & \mathbf{Q}_j(U^\kappa) \end{array}\right] = r_i + \operatorname{rank}\ \mathbf{Q}_j(U^\kappa). \tag{5}$$

In this case, the subscheme S_j parameters can be unidentifiable due to the fact that $\operatorname{rank}\mathbf{Q}_j(U^\kappa) < r_j$.

Let us write down the necessary condition for the distinguish ability of the parameters of the subschemes S_i, S_j, under the hypothesis H_{ρ_i}

$$\operatorname{rank}\left[\mathbf{Q}_i(U^\kappa)\middle|\mathbf{Q}_{j,s}(U^\kappa)\right] = r_i + 1;\ S = \overline{1, r_j},$$

where $\mathbf{Q}_{j,s}(U^\kappa)$ — is the s-th column of the matrix $\mathbf{Q}_j$.

If only one parameter in the system is allowed to change, then condition (5) with respect to the distinguish ability of all the system parameters reduces to the existence of input signals of the system ensuring pair wise linear independence of the columns of the matrix $\left[\mathbf{Q}_1(U^\kappa),\ldots,\mathbf{Q}_N(U^\kappa)\right]$, where N is the total number of parameters tested [5]. And linear independence of different columns can be achieved with different input signals of the system.

1. Planning a diagnostic experiment when estimating the subscheme parameters. Let the scalar output of y be measured with an error. Assuming that only parametric faults can exist in the subscheme S_i and the system model is linear in the parameters of the subsystems under test, the description of the faulty system is similar to (2) in the form $\Delta y + \xi = q_i(u)\Delta\rho_i$. Changing the values of the input signals of the system, we form, similarly to (3), a system of algebraic equations for obtaining estimates $\Delta\hat{\rho}_i$ of the parameters $\Delta\rho_i$:

$$\Delta\Lambda = \mathbf{Q}_i\ \Delta\rho_i, \tag{6}$$

where $\Delta\Lambda = \left[\Delta y\left(u^1\right) + \xi^1,\ldots,\Delta y\left(u^\kappa\right) + \xi^\kappa\right]^T$, $\mathbf{Q}_i = \mathbf{Q}_i\left(U^\kappa\right)$.

Suppose that the measurement error values ξ^r have a normal distribution with zero expectation $E\left[\xi^r\right] = 0$ and the same variance σ^2, are not correlated with each other, and also with $q_i(u^r)$, $\Delta\rho_i$. In this case, the quantities $\Delta y(u^r) + \xi^r$, $r = \overline{1, K}$ are independent, normally distributed, with mathematical expectation $\Delta y\left(u^r\right)$ and the same variance σ^2.

The assumptions made allow us to use the method of least squares to obtain an estimate $\Delta\hat{\rho}_i$:

$$\Delta\hat{\rho}_i = \left(\mathbf{Q}_i^{\mathrm{T}}\mathbf{Q}_i\right)^{-1}\mathbf{Q}_i^{\mathrm{T}}\ \Delta\Lambda. \tag{7}$$

It is assumed that the matrix $\mathbf{Q}_i^{\mathrm{T}}\mathbf{Q}_i = \mathbf{F}_i$, called the information matrix (the Fisher matrix), is no degenerate.

If the vector were measured without errors, then from (3) it would be possible to obtain exact (errors of computation are not considered) parameter values $\Delta\rho_i = \left(\mathbf{Q}_i^{\mathrm{T}}\mathrm{Q}_i\right)^{-1}\mathbf{Q}_i^{\mathrm{T}}\Delta Y$. The covariance matrix of estimates of the subscheme S_i parameters is as follows:

$$\mathbf{cov}\left[\Delta\hat{\rho}_i\right] = \mathbf{E}\left[(\Delta\hat{\rho}_i - \Delta\rho_i)(\Delta\hat{\rho}_i - \Delta\rho_i)^{\mathrm{T}}\right] = \sigma^2\ \mathbf{F}_i.$$

Criteria for the optimal choice of input signals of the system are related to the form of the information matrix $\mathbf{F}_i$ and are aimed at minimizing any of its characteristics, for example, the value of the determinant [6].

The foregoing experimental planning assumes that the faulty subscheme is known and it is only necessary to determine its parameters. If the faulty subscheme is unknown, then when evaluating the parameters of the scanned subscheme, it is necessary to ensure that the parameters of the subschema are discernible with the subscheme parameters, which can in fact be faulty.

2. Planning a diagnostic experiment for parametric faults. Consider the planning of an experiment aimed at ensuring the distinguish ability of the parameters of subsystems S_i, S_j, when estimating the parameters of the subsystem S_i.

We divide the diagnostic equation (6) into two parts:

$$\mathbf{\Delta\Lambda}_{\mathrm{a}} = \mathbf{Q}_{i,\mathrm{a}}\,\mathbf{\Delta}\rho_i, \tag{8}$$

$$\mathbf{\Delta\Lambda}_{\mathrm{b}} = \mathbf{Q}_{i,\mathrm{b}}\,\mathbf{\Delta}\rho_i. \tag{9}$$

The compatibility of equation (6) can be checked by checking the unbiasedness of the estimates $\mathbf{\Delta}\hat{\rho}_i^{\mathrm{a}}$, $\mathbf{\Delta}\hat{\rho}_i^{\mathrm{b}}$, the parameters $\mathbf{\Delta}\rho_i$ obtained from equations (8) and (9), respectively. Analogously to (7) from (8) we obtain an estimate of the parameters

$$\mathbf{\Delta}\hat{\rho}_i^{\mathrm{a}} = \left(\mathbf{Q}_{i,\mathrm{a}}^{\mathrm{T}}\,\mathbf{Q}_{i,\mathrm{a}}\right)^{-1}\mathbf{Q}_{i,\mathrm{a}}^{\mathrm{T}}\,\mathbf{\Delta\Lambda}_{\mathrm{a}} = \mathbf{F}_{i,\mathrm{a}}\,\mathbf{Q}_{i,\mathrm{a}}^{\mathrm{T}}\,\mathbf{\Delta\Lambda}_{\mathrm{a}}.$$

From (9) we obtain $\mathbf{\Delta}\hat{\rho}_i^{\mathrm{b}} = \mathbf{F}_{i,\mathrm{b}}\,\mathbf{Q}_{i,\mathrm{b}}^{\mathrm{T}}\,\mathbf{\Delta\Lambda}_{\mathrm{b}}$.

If the subscheme S_j is faulty, the matrices $\mathbf{\Delta\Lambda}_{\mathrm{a}}$, $\mathbf{\Delta\Lambda}_{\mathrm{b}}$ are determined by expressions $\mathbf{\Delta\Lambda}_{\mathrm{a}} = \mathbf{Q}_{j,\mathrm{a}}\,\mathbf{\Delta}\rho_j$, $\mathbf{\Delta\Lambda}_{\mathrm{b}} = \mathbf{Q}_{j,\mathrm{b}}\,\mathbf{\Delta}\rho_j$.

Because of measurement errors $\mathbf{\Delta}\hat{\rho}_i^{\mathrm{a}}$, $\mathbf{\Delta}\hat{\rho}_i^{\mathrm{b}}$, estimates are random quantities with some areas of distribution of their values.

Depending on which of the subscheme (S_i or S_j) is faulty, the mathematical expectation $E[\mathbf{\Delta}\hat{\rho}_i^{\mathrm{a}}]$ and the covariance matrix $\mathbf{cov}[\mathbf{\Delta}\hat{\rho}_i^{\mathrm{a}}]$ of the estimate $\mathbf{\Delta}\hat{\rho}_i^{\mathrm{a}}$ obtained under the hypothesis H_{ρ_i} are as follows:

$$E[\mathbf{\Delta}\hat{\rho}_{\mathrm{i}}^{\mathrm{a}}] = \begin{cases} \mathbf{F}_{i,\mathrm{a}}\,\mathbf{Q}_{j,\mathrm{a}}^{\mathrm{T}}\,\Delta\rho_j & \text{at } j \neq i, \\ \mathbf{\Delta}\rho_i & \text{at } j = i, \end{cases}$$

$\mathbf{cov}[\mathbf{\Delta}\hat{\rho}_i^{\mathrm{a}} = \sigma^2\mathbf{F}_{i,\mathrm{a}}] = \sigma^2\,\mathbf{F}_{i,\mathrm{a}}$ — if any of the subschemes is faulty. Similar expressions are also obtained for $E\left[\mathbf{\Delta}\hat{\rho}_i^{\mathrm{b}}\right]$, $\mathbf{cov}\left[\mathbf{\Delta}\hat{\rho}_i^{\mathrm{b}}\right]$, by substituting in the expressions for $E\left[\mathbf{\Delta}\hat{\rho}_i^{\mathrm{a}}\right]$, $\mathbf{cov}\left[\mathbf{\Delta}\hat{\rho}_i^{\mathrm{a}}\right]$, the index «a» by «b».

The bias in the estimate $\mathbf{\Delta}\hat{\rho}_i$ is determined by the magnitude

$$\boldsymbol{\rho}_i = \mathbf{\Delta}\hat{\rho}_i^{\mathrm{a}} - \mathbf{\Delta}\hat{\rho}_i^{\mathrm{b}}. \tag{10}$$

The mathematical expectation and the covariance matrix of the vector $\boldsymbol{\rho}_i$ are as follows:

$$E\left[\boldsymbol{\rho}_i\right] = \begin{cases} \mathbf{m}_{i,i} = \left(\mathbf{F}_{i,\mathrm{a}}\,\mathbf{Q}_{i,\mathrm{a}}^{\mathrm{T}}\,\mathbf{Q}_{j,\mathrm{a}} - \mathbf{F}_{i,\mathrm{b}}\,\mathbf{Q}_{i,\mathrm{b}}^{\mathrm{T}}\,\mathbf{Q}_{j,\mathrm{b}}\right)\mathbf{\Delta}\rho_j & \text{at } j \neq i, \\ \mathbf{m}_{i,j} = 0 & \text{at } i = j, \end{cases} \tag{11}$$

$$\mathbf{C}_i = \mathbf{cov}[\rho_{\mathrm{i}}] = \sigma^2(\mathbf{F}_{i,\mathrm{a}} + \mathbf{F}_{i,\mathrm{b}}). \tag{12}$$

As follows from (12), the covariance matrix of the vector $\boldsymbol{\rho}_{\mathrm{i}}$ does not depend on which of the subschemes is faulty.

Let the system change only one parameter, i.e. $\Delta\rho_i$ — is a scalar. If the system of diagnostic equations (8), (9) consists of two scalar equations

$$\Delta\Lambda_{i,\mathrm{a}} = q_{i,\mathrm{a}}\,\Delta\rho_i,$$

$$\Delta\Lambda_{i,\mathrm{b}} = q_{i,\mathrm{b}}\,\Delta\rho_i,$$

then

$$E\left[\rho_i\right] = \begin{cases} m_{i,i} = \left(\dfrac{q_{j,\mathrm{a}}}{q_{i,\mathrm{a}}} - \dfrac{q_{j,\mathrm{b}}}{q_{i,\mathrm{b}}}\right)\Delta\rho_j & \text{at } j \neq i, \\ m_{i,j} = 0 & \text{at } j = i, \end{cases}$$

$$\beta_i = D[\rho_i] = \delta^2\left(\frac{1}{q_{i,\mathrm{a}}^2} + \frac{1}{q_{i,\mathrm{b}}^2}\right).$$

For this distinction of subschemes S_i, S_j, it is necessary that the distribution areas of the quantities $\rho_{i,i}$ (ρ_i with a faulty subscheme S_i) and $\rho_{i,j}$ (ρ_j with a faulty subscheme S_j) do not overlap.

As an example, Fig. 1 shows the distribution of two-dimensional quantities $\rho_{i,i}$, $\rho_{i,j}$. The situation shown in Fig. 1, a, is characterized with respect to the situation shown in Fig. 1, b, higher accuracy of parameter estimates $\Delta\hat{\rho}_i$, but lower distinguish ability of subschemes S_i, S_j.

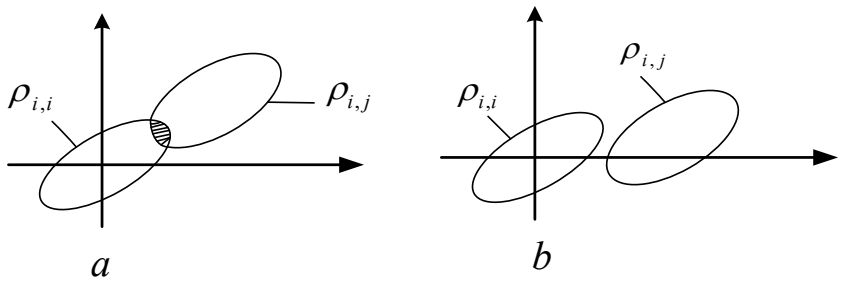


Fig.1. — Areas of distribution of two-dimensional quantities $\rho_{i,i}$, $\rho_{i,j}$.

In the present paper, the problem of determining the optimal input signals of a system in the planning of a diagnostic experiment that ensure the distinguish ability of subschemes S_i, S_j is considered as the problem of obtaining the maximum distance between the regions of distribution of the values of $\rho_{i,i}$ and $\rho_{i,j}$.

In the theory of pattern recognition, the distance of the Mahalanobis is widely used as a measure of the distance between sets [7]. Assume that the distribution of values $\rho_{i,i}$, $\rho_{i,j}$, obey the normal laws $N(\mathbf{m}_{i,i}, \mathbf{C}_i)$, $N(\mathbf{m}_{i,j}, \mathbf{C}_i)$, where $\mathbf{m}_{i,i}$, $\mathbf{m}_{i,j}$ — are the vectors of mathematical expectation, $\mathbf{C}_i$ — is the covariance matrix of dimension. In this case, the generalized Mahalanobis distance between the regions of distribution $\rho_{i,i}$ and $\rho_{i,j}$ is determined by the expression

$$\mathbf{I}_{i,j} = \mathbf{M}_{i,j}^{\mathrm{T}} \mathbf{C}_i^{-1} \mathbf{M}_{i,j}, \tag{13}$$

then $\mathbf{M}_{i,j} = \mathbf{m}_{i,i} - \mathbf{m}_{i,j}$.

If the covariance matrix $\mathbf{C}_i$ — is a single matrix, then $\mathbf{I}_{i,j}$ it characterizes the quadratic distance between the mathematical expectations $\rho_{i,i}$ and $\rho_{i,j}$.

For one-dimensional normally distributed sets of values $\rho_{i,i}$ and $\rho_{i,j}$, expression (13) takes the form

$$\mathbf{I}_{i,j} = \frac{(\mathbf{m}_{i,i} - \mathbf{m}_{i,j})^2}{\beta_i}. \tag{14}$$

Since according to (11) $\mathbf{m}_{i,i} = \mathbf{0}$, then (13), (14) can be written accordingly:

$$\mathbf{I}_{i,j} = \mathbf{m}_{i,j} \mathbf{C}_i^{-1} \mathbf{m}_{i,j}, \tag{15}$$

$$\mathbf{I}_{i,j} = \frac{\mathbf{m}_{i,j}^2}{\beta_i}. \tag{16}$$

For calculations using formulas (15), (16), it is necessary to know the value $\mathbf{m}_{ij}$, which, according to (12), depends on the unknown value $\mathbf{\Delta}\rho_j$.

If $\Delta\rho_j$ — is a scalar, i.e. only one parameter in the system is allowed to change, the maximum value I_{ij} obtained for an arbitrary value $\Delta\rho_j \neq 0$ of for $u = u^0$ is the maximum (but different in magnitude) for any value $\Delta\rho_j$. It follows from this that the optimal input

signals u^0, at which the best distinguish ability of the subschemes S_i, S_j is achieved, can be determined before the diagnostic experiment at an arbitrary value $\Delta\rho_j$ by maximizing the value I_{ij}.

If $\Delta\rho_j$ — is a vector, then it is impossible to obtain optimal values u^0 beforehand before the diagnostic experiment, delivering a maximum $\mathbf{I}_{ij}$ for any values $\mathbf{\Delta}\rho_j$. This is due to the fact that the value $\mathbf{m}_{ij}$ depends on the combination of the values of the components of the vector $\mathbf{\Delta}\rho_j$. In this case, the choice of input signals aimed at ensuring the distinguish ability of subschemes S_i, S_j can be carried out during the diagnostic experiment. Consider one of the variants of the experiment, consisting of two stages.

At the *first stage*, the estimates of the parameters of all subsystems are determined successively $\Delta\hat{\rho}_1,\ldots,\Delta\hat{\rho}_N$. The input signals of the system at this stage are selected from the conditions of optimal experiment planning for identifying the parameters of the corresponding subsystem. Among the estimates obtained $\Delta\hat{\rho}_1,\ldots,\Delta\hat{\rho}_N$, only one corresponds to the true values of the parameters of the faulty subsystem.

At the *second stage*, the optimal input signals of the system are determined, allowing selection of the obtained estimates $\Delta\hat{\rho}_1,\ldots,\Delta\hat{\rho}_N$ under the appropriate hypothesis. In this case, the criterion of optimality under the hypothesis H_{ρ_i} is the maximum of the quantity calculated by formula (13). In calculations according to formula (11), an estimate from $\Delta\hat{\rho}_1,\ldots,\Delta\hat{\rho}_N$ the set corresponding to the subsystem is used, with respect to which the distinguishable subscheme S_i is distinguishable.

Use to test the hypothesis H_{ρ_i} of the magnitude of the bias of the estimate $\Delta\hat{\rho}_i$, determined by the expression (10), is possible only if for matrices $\mathbf{Q}_{i,\mathrm{a}}$, $\mathbf{Q}_{i,\mathrm{b}}$, there are minors of rank r_i. If for one of the matrices, for example $\mathbf{Q}_{i,\mathrm{b}}$, the matrix does not have a minor of rank r_i, then the reliability of the estimate $\mathbf{\Delta}\hat{\rho}_i^{\mathrm{a}}$ can be checked by the criterion of the forecast of the output signal of the system.

In this case, an estimate $\mathbf{\Delta}\hat{\rho}_i = \mathbf{F}_{i,\mathrm{a}}\,\mathbf{Q}_{i,\mathrm{a}}^{\mathrm{T}}\,\mathbf{\Delta\Lambda}_{\mathrm{a}}$ is determined from (7), and then, according to (10), the quantity $\mathbf{\Delta}\hat{\Lambda}_{\mathrm{b}} = \mathbf{Q}_{i,\mathrm{b}}\,\mathbf{\Delta}\hat{\rho}_{\mathrm{i}}$.

Error forecast

$$\boldsymbol{\mu}_i = \mathbf{\Delta\Lambda}_{\mathrm{b}} - \mathbf{\Delta}\hat{\Lambda}_{\mathrm{b}} \tag{17}$$

is an indicator of the correctness of the hypothesis H_{ρ_i}. If $\boldsymbol{\mu}_i < \boldsymbol{\varepsilon}_i$ that hypothesis H_{ρ_i} is accepted, otherwise - it is rejected ($\boldsymbol{\varepsilon}_i$ — an acceptable value $\boldsymbol{\mu}_i$).

The value $\boldsymbol{\mu}_i$ depends on which subsystem is actually faulty. If we consider subschemes S_i, S_j, then the mathematical expectation and variance of the quantities $\boldsymbol{\mu}_i$ are determined by the expressions

$$E[\boldsymbol{\mu}_i] = \begin{cases} \boldsymbol{\mu}_{ij} = \mathbf{Q}_{i,\mathrm{b}}\,\mathbf{F}_{i,\mathrm{a}}\,\mathbf{Q}_{i,\mathrm{a}}^{\mathrm{T}}\,\mathbf{\Delta}\rho_j & \text{at } j \neq i, \\ \boldsymbol{\mu}_{ij} = \mathbf{0} & \text{at } j = i, \end{cases} \tag{18}$$

$$\mathbf{cov}[\boldsymbol{\mu}_i] = \sigma^2\left(\mathbf{1} + \mathbf{Q}_{i,\mathrm{b}}^{\mathrm{T}}\,\mathbf{F}_{i,\mathrm{a}}\,\mathbf{Q}_{i,\mathrm{b}}\right), \tag{19}$$

where $\mathbf{1}$ — is the unit diagonal matrix.

The criterion of optimality of input signals for a hypothesis H_{ρ_i} oriented to the determination of a defective subschemes S_i, S_j in a pair can be the maximum of (15), where

$\mathbf{m}_{ij}$ it should be put equal $\boldsymbol{\mu}_{ij}$, and $\mathbf{C}_i = \mathbf{cov}[\boldsymbol{\mu}_i]$. Otherwise, for the criterion based on the expression (18), the reasoning given for the criterion based on the expression (10) can be repeated.

3. Planning a diagnostic experiment for structural faults. Let us consider subsystems with *independent observation*. When testing the hypothesis $H_i : \Delta y = L_i(u)\Delta Z_i$, $i = \overline{1,N}$ (where L_i — is the class of operators that specifies a class of faults in the subscheme S_i [1]), the compatibility of equations of the form is verified:

$$\boldsymbol{\Delta}\mathrm{Y}_\mathrm{a} + \boldsymbol{\xi}_\mathrm{a} = \mathbf{L}_{i,\mathrm{a}}\,\boldsymbol{\Delta Z}_i, \tag{20}$$

$$\boldsymbol{\Delta}\mathrm{Y}_\mathrm{b} + \boldsymbol{\xi}_\mathrm{b} = \mathbf{L}_{i,\mathrm{b}}\,\boldsymbol{\Delta Z}_i. \tag{21}$$

In this case $\boldsymbol{\Delta}\mathrm{Y}_\mathrm{a}$, $\boldsymbol{\Delta}\mathrm{Y}_\mathrm{b}$, as well $\boldsymbol{\xi}_a$, $\boldsymbol{\xi}_\mathrm{b}$, are vectors whose dimensions are equal to or greater than the dimension of the vector $\boldsymbol{\Delta}\mathrm{Z}_\mathrm{i}$.

Let the decision on the reliability of the hypothesis H_i be made by the criterion of unbiasedness of the estimate $\boldsymbol{\Delta}\hat{\mathrm{Z}}_i$. Offsetting of the estimate $\boldsymbol{\Delta}\hat{\mathrm{Z}}_i$ is similar to (10) determined by the quantity

$$\boldsymbol{\tau}_\mathrm{i} = \boldsymbol{\Delta}\hat{Z}_i^\mathrm{a} - \boldsymbol{\Delta}\hat{Z}_i^\mathrm{b}. \tag{22}$$

If the matrix $\mathbf{L}_{i,\mathrm{a}}$ is rectangular and the method of least squares is used to obtain the estimate $\boldsymbol{\Delta}\hat{\mathrm{Z}}_i^\mathrm{a}$ of the vector $\boldsymbol{\Delta}\mathrm{Z}_i$ from equation (20), then

$$E\left[\boldsymbol{\Delta}\hat{Z}_\mathrm{i}^\mathrm{a}\right] = \begin{cases} \boldsymbol{\theta}_{i,\mathrm{a}}\,\mathbf{L}_{i,\mathrm{a}}^\mathrm{T}\,\mathbf{L}_{j,\mathrm{a}}\,\boldsymbol{\Delta Z}_j & \text{at } j \neq i, \\ \boldsymbol{\Delta Z}_\mathrm{i} & \text{at } j = i, \end{cases}$$

then $\boldsymbol{\theta}_{i,\mathrm{a}} = \left(\mathbf{L}_{i,\mathrm{a}}^\mathrm{T}\,\mathbf{L}_{i,\mathrm{a}}\right)^{-1}$.

The estimate $\boldsymbol{\Delta}\hat{\mathrm{Z}}_i^\mathrm{b}$ of the vector $\boldsymbol{\Delta Z}_i^\mathrm{b}$ from equation (21) is determined in a similar way.

The value $\boldsymbol{\tau}_i$ determined by the hypothesis H_i depends on which subscheme is actually faulty. Consider the case where only one of the subschemes S_i, S_j. can be faulty.

Analogously to (11), (12) we obtain

$$E[\boldsymbol{\tau}_i] = \begin{cases} \boldsymbol{\tau}_{i,j} = (\boldsymbol{\theta}_{i,\mathrm{a}}\,\mathbf{L}_{i,\mathrm{a}}^\mathrm{T}\,\mathbf{L}_{j,\mathrm{a}} - \boldsymbol{\theta}_{i,\mathrm{b}}\,\mathbf{L}_{i,\mathrm{b}}^\mathrm{T}\,\mathbf{L}_{j,\mathrm{b}})\boldsymbol{\Delta}Z_j & \text{at } j \neq i, \\ \boldsymbol{\tau}_{i,i} = \boldsymbol{\theta} & \text{at } j = i, \end{cases} \tag{23}$$

$$\mathbf{cov}[\boldsymbol{\tau}] = \sigma^2(\boldsymbol{\theta}_{i,\mathrm{a}} + \boldsymbol{\theta}_{i,\mathrm{b}}). \tag{24}$$

For the case when ΔZ_i, ΔZ_j — are scalar quantities, that is, subschemes S_i, S_j, have one output, we get

$$E[\tau_i] = \begin{cases} \tau_{i,j} = \left(\dfrac{L_{j,\mathrm{a}}}{L_{i,\mathrm{a}}} - \dfrac{L_{j,\mathrm{b}}}{L_{i,\mathrm{b}}}\right)\Delta\, Z_j & \text{at } j \neq i, \\ \tau_{i,i} = 0 & \text{at } j = i, \end{cases}$$

$$D[\tau_i] = \sigma^2\left(\frac{1}{L_{i,\mathrm{a}}^2} + \frac{1}{L_{i,\mathrm{b}}^2}\right).$$

The principal difference between experiment planning and *structural faults* from planning for *parametric faults* $\boldsymbol{\Delta}\rho_i$ is that for parametric faults the value does not depend on the input signals of the system, while for structural faults the value $\boldsymbol{\Delta}\rho_i$ depends on the input signals of the system, in an unknown way.

Let the observability matrices $\mathbf{L}_{i,\text{a}}$, $\mathbf{L}_{i,\delta}$, $\mathbf{L}_{j,\text{a}}$, ,$\mathbf{L}_{j,\delta}$ — be constants and ΔZ_j — be a scalar quantity. In this case, when organizing a diagnostic experiment, two stages can be distinguished.

The first stage of the connection with the formation of observability matrices before the diagnostic experiment by means of an appropriate selection of control points. The choice of control points is carried out from the condition of maximum value (21), where $\mathbf{m}_{i,j}$ it is replaced by $\boldsymbol{\tau}_{i,j}$, and $\mathbf{C}_i$ by $\mathbf{cov}[\boldsymbol{\tau}_i]$. Since the maximum value $\mathbf{I}_{i,j}$ obtained for the corresponding set of control points for arbitrary values $\Delta\mathbf{Z}_j \neq \mathbf{0}$ remains the maximum (but different in magnitude) for any values $\Delta\mathbf{Z}_j$, the choice of control points can be carried out at arbitrary values $\Delta\mathbf{Z}_j \neq \mathbf{0}$.

The second stage is realized during the diagnostic experiment. Since the observability matrices are constant in this case, no increase in noise occurs when the input signals of the system change, which follows from (24). With a faulty subscheme S_j, we have $\Delta y_{\text{a}} = \mathbf{L}_{j,\text{a}}\,\Delta Z_j$, $\Delta y_{\text{b}} = \mathbf{L}_{j,\text{b}}\,\Delta Z_j$. In this case, the quantity $\boldsymbol{\tau}_{i,j}$ in expression (23) corresponds to the quantity

$$\boldsymbol{\tau}^{*}_{i,j} = \boldsymbol{\theta}_{i,\text{a}}\,\mathbf{L}^{\text{T}}_{i,\text{a}}\,\Delta y_{\text{a}} - \boldsymbol{\theta}_{i,\text{b}}\,\mathbf{L}^{\text{T}}_{i,\text{b}}\,\Delta y_{\text{b}}.$$

The input signals of the system that ensure the best distinguish ability of the subschemes S_i, S_j under the hypothesis H_i are chosen in the second stage from the condition of the maximum of the quantity (21), where $\mathbf{m}_{i,j}$ it is replaced by $\boldsymbol{\tau}^{*}_{i,j}$, and $\mathbf{C}_i$ — by $\mathbf{cov}[\boldsymbol{\tau}_i]$.

When choosing control points, a table of coverings is usually constructed, the columns of which correspond to pairs of subsystems whose distinguishability is estimated, and the rows to checkpoints. At the intersection of a row and a column, there is one if some pair of subsystems are distinguishable by the introduction of the corresponding control point. The minimum set of control points corresponds to the minimum coverage of the table. There can be several minimal coverages, the choice of which is not formalized.

In some cases, for example, as described in this section, instead of one in the table of coverings, one can record the corresponding value of the Mahalanobis distance and from the minimal coverages choose a cover that provides a maximum, in some sense, distinguish ability of the subsystems. The criterion for choosing such a minimum coverage can be, in particular, the maximum value of the sum of all elements of the minimal coverage table.

If there is a limit on the number of channels of signal transmission from control points to the diagnostic system, the diagnostic system can be equipped with a diagnostic surveillance system. The diagnostic monitoring system is an adder whose inputs are connected to control points, and the output is to a communication channel. The weighting factors of the adder are chosen from the conditions for maximum discrimination of the subsystems on the basis of the Mahalanobis distance analysis for given ones.

Conclusions. For cases of parametric and structural faults obtained evaluation to test hypotheses about the distinctiveness of faulty subschemes in freewheeling systems. These estimates allow to identify faulty subscheme and plan their computational experiments based on the analysis of localization parameters subschemes diagnosable system to the test exposure. In this case, hypotheses are formulated from the assumption that the parameters of the diagnosed subscheme have changed. The localization of faulty system subschemes during the diagnostic experiment is carried out from the assumption of the distinguish ability of the subschemes, i.e. the partitioning of the system into subschemes must be carried out in such a way that the regions of distribution of the parameters of the subschemes do not intersect.

Referents

1. Верлань А.Ф., Положаєнко С.А. Локалізація несправних фрагментів при діагностуванні безінерційних систем. Електротехнічні та комп'ютерні системи. Теорія і практика. Спеціальний випуск. Астропрінт. 2017. С. 439–445.
2. Верлань А.А., Осман І.Х., Положаєнко С.А. Декомпозиційний метод локалізації несправних електронних підсхем. Електромашинобудування та електрообладнання, 2007. Вип. 69. С. 72–76.
3. Верлань А.А., Стертен Ю.І., Положаєнко С.А. Формалізація представлення послідовності тестових гіпотез при діагностуванні електронних схем. Інформатика та математичні методи в моделюванні. 2016. Vol. 6, № 4. P. 315–321.
4. Положаєнко С.А., Верлань А.А., Осман І.Х. Локалізація несправних електронних підсхем методом навчаючих та перевірочних характеристик. Математичне та комп'ютерне моделювання. Серія: Технічні науки: зб. наук. пр. Кам'янець-Подільський: Кам'янець-Подільськ. нац. ун-т, 2008. Вип. 1. С. 140–144.
5. Savelev A.A., Prokofieva L.L. Methods of Mathematical Simulation and Machine Identification of Abnormal Energy Processes [Text]. Colloquium-journal. Part 1, 2024. № 16 (209). P. 36–42. https://doi.org/10.24412/2520-6990-2024-16209-36-42.
6. Verlan A.A., Plozhaenko S., Prokofieva L., Shilov V. Algorithmization of the failed subschemes localization process. Herald of advanced information technology. 2019. №1 (02). P. 37- 46. DOI://10.15276/hait.02.2019.4
7. Прокофьєва Л.Л., Савєльєв А.А. Евристичні моделі вимірювальних процедур в задачах аналітичного дослідження тензометричних систем. Математичне та комп'ютерне моделювання. Серія: Технічні науки: зб. наук. пр. Кам'янець-Подільський: Кам'янець-Подільськ. нац. ун-т, 2023. Вип. 24. С. 56–67. DOI: 10.32626/2308-5916.2023-24.67-78 URL: http://mcm-tech.kpnu.edu.ua/issue/archive

ПЛАНУВАННЯ ДІАГНОСТИЧНОГО ЕКСПЕРИМЕНТУ ПРИ ЛОКАЛІЗАЦІЇ НЕСПРАВНОСТЕЙ ПІДСХЕМ БЕЗІНЕРЦІЙНИХ СИСТЕМ

А.А. Савельєв, Л.Л. Прокофьєва

Національний університет «Одеська політехніка»
1, Шевченка пр., м.Одеса, 65044, Україна
Emails: saveleva@op.edu.ua, prokofieva1957@gmail.com

Отримано формалізовані умови проведення діагностичного експерименту. пов'язаного з виявленням несправних фрагментів (підсхем) безінерційних систем. Діагностичний експеримент зведено при цьому до обчислювальних процедур локалізації несправних підсхем, в основу яких (процедур) покладено перевірку гіпотез про те, що змінилися характеристики виділених підсхем. Гіпотези формулюються таким чином, щоб забезпечити виявлення параметричних та структурних несправностей. До перших, наприклад, можуть відноситися зміни опору ділянки ланцюга , а до других — обрив або коротке замикання. При розбитті системи на підсхеми, останні обираються за умови можливості їх параметричної ідентифікації. Тобто ситуації, коли по відомих параметрах інших підсхем, а також вхідних та вихідних сигналах системи в цілому, можна визначити параметри підсхеми, яка розглядається. Планування діагностичного експерименту полягає у наступному. Передбачаючи лінійну залежність між параметрами виділеної підсхеми S_i та вихідними сигналами системи, завчасно складається перевірочна матриця $\mathbf{F}$, яка визначає взаємозв'язок вказаних параметрів та сигналів, враховуючи справність підсхеми, що розглядається. При виникненні несправностей перевірочна матриця $\mathbf{F}$ змінюється, що дозволяє визначити матрицю невідповідності $\mathbf{q}$, на підставі аналізу якої можна отримати оцінку $\Delta\hat{\rho}_i$ параметрів $\Delta\rho_i$ підсхеми S_i, яка розглядається. В результаті даного аналізу робиться висновок щодо працездатності підсхеми S_i. Процедури перевірки гіпотез щодо справності підсхем без інерційної системи для випадків параметричних та структурних несправностей мають ідентичний характер. Принципова відмінність планування експериментів при структурних несправностях від планування при параметричних несправностях полягає у тому, що при параметричних несправностях величина $\Delta\rho_i$ не залежить від вхідних сигналів системи, а при структурних — величина $\Delta\rho_i$ залежить від вхідних сигналів системи, причому невідомим чином.

Ключові слова: діагностика, діагностичний експеримент, локалізація несправностей, безінерційні системи, оцінювання параметрів підсхем.

DOI 10.15276/imms.v15.no2.176
УДК 004.738.5:351

Informatics and Mathematical Methods in Simulation
Vol.15 (2025), No. 2, pp. 176-186

АЛГОРИТМИ ТА МЕТОДИ МОДЕЛЮВАННЯ ІНФОРМАЦІЙНИХ ОПЕРАЦІЙ У СОЦІАЛЬНИХ МЕРЕЖАХ

О.О. Васильєва

Національний університет «Чернігівська політехніка»
95 Шевченка, вул., м. Чернігів, 14030, Україна
Email: olga.vasiljeva37@gmail.com

У статті проведено аналіз сучасних наукових експериментів, що присвячені темі моделювання інформаційних операцій в соціальних мережах кіберпростору, який поглиблює розуміння актуальність підходів та їх інтеграцію з сучасними технологіями. Проведено загальну оцінку проаналізованих підходів та акцентовано увагу на зростаючій ролі штучного інтелекту, мовних моделей та адаптивної динаміки мереж зумовлює необхідність створення нових агентно-орієнтованих архітектур. Наведено поняття інформаційних операцій, здійснено порівняння із сумісними поняттями. Орієнтуючись на основні властивості соціальних мереж – мультимедійності, інтерактивності, можливості зворотного зв'язку, горизонтальності зв'язків, відсутності посередників, позагеографічності або позапросторовості – запропоновано архітектуру та логіку функціонування інформаційної операції в соціальній мережі. Графічно представлено загальний алгоритм інформаційної операції в соціальних мережах кіберпростору, що включає ідентифікацію ключових агентів, прийняття рішень цими агентами, виявлення спільнот та кластерів агентів. Розглянуто класи методів та алгоритмів, що застосовуються для моделювання інформаційних операцій у соціальних мережах. Запропоновано комбінації методів з точки зору агентного моделювання, що поєднують Алгоритми поширення впливу + Алгоритми ухвалення рішень; Еволюційні алгоритми + Методи мережевого аналізу; Методи кластеризації + Методи динамічного моделювання; Методи обробки великих даних + Агентне моделювання. Для побудови агентної моделі, спрямованої на моделювання інформаційних операцій у соціальних мережах, запропоновано поєднання одразу п'яти методів у визначеній послідовності з урахуванням їхньої інтеграції в загальну агентну модель, та наведено приклад сценарію комплексного моделювання інформаційних операцій в соціальних мережах кіберпростору.

Ключові слова: інформаційні операції, соціальні мережі, агентне моделювання, еволюційні алгоритми, теорія ігор, машинне навчання, MCMC, Louvain, Influence Maximization.

Вступ. У сучасних умовах гібридної війни, інформаційні операції у соціальних мережах стали одним із ключових інструментів впливу на громадську думку, маніпуляції поведінкою мас та послаблення інформаційної безпеки держав. У відповідь на зростання масштабів і складності інформаційних операцій з'являється необхідність у створенні точних і гнучких моделей, які дозволяють виявляти, прогнозувати та протидіяти цим загрозам. Агентне моделювання як метод дослідження складних соціально-інформаційних систем демонструє високу ефективність у відтворенні динаміки поширення інформації та стратегічної взаємодії між учасниками соціальних мереж.

Розробка відповідних алгоритмічних інструментів для моделювання інформаційних операцій вимагає залучення міждисциплінарних підходів: від теорії ігор і еволюційної оптимізації до глибокого навчання та мережевого аналізу. Особливу актуальність набуває можливість комбінування методів у межах агентно-орієнтованих моделей, що дозволяє не лише аналізувати поведінку окремих агентів, але й оцінювати макроефекти у межах всієї інформаційної екосистеми.

Мета дослідження. Метою цієї статті є узагальнення та класифікація методів і алгоритмів, що використовуються для моделювання інформаційних операцій у соціальних мережах на основі агентного підходу, а також побудова ефективної

комбінації цих методів для створення комплексної симуляційної моделі. Особлива увага приділяється інтеграції алгоритмів поширення впливу, машинного навчання, методів динамічного моделювання та оптимізації поведінки агентів. Результати мають стати основою для розробки прикладних рішень у сфері інформаційної безпеки, контрдезінформаційних стратегій та аналізу соціальної динаміки.

Аналіз останніх досліджень і публікацій. Аналіз публікацій та наукових експериментів, проведений автором, вказує на значний інтерес наукової спільноти до теми та підтверджує її актуальність. Так Iamnitchi et al. (2023) [1] у межах ініціативи DARPA SocialSim представили модель моделювання інформаційних потоків у соціальних мережах, що враховує міжплатформну динаміку, поведінкову еволюцію користувачів та реактивну дезінформацію. Автори дійшли висновку щодо необхідності інтеграції поведінкових моделей та часових закономірностей у механізми симуляції впливу. Порівняльний огляд процесів дифузії інформації, класифікуючи підходи за типами моделей, джерелами даних та сферою застосування провели Islam et al. (2024) [2]. Автори підкреслюють ефективність поєднання мережевих та когнітивних моделей для вивчення складних інформаційних процесів. Liao et al. (2024) [3] представили у журналі *Scientific Reports* моделі поширення дезінформації як епідемічний процес із врахуванням тональності контенту та часових інтервалів втручання. Вони доводять, що вчасна інтервенція є критичною для зменшення охоплення деструктивних меседжів. Gao et al. (2024) [4] демонструють можливість використання великих мовних моделей (LLM) для імітації соціальної поведінки агентів у багатоканальних цифрових середовищах. Дослідження виявило, що агенти на основі LLM здатні генерувати когерентні стратегії впливу в залежності від контексту. Ding et al. (2024) інтегрують генеративний ШІ у агентно-орієнтовані моделі, де агенти симулюють поведінку користувачів Twitter у контексті інформаційних операцій [5]. Моделі відображають як мовні патерни, так і структури мережевої взаємодії. Kong et al. (2022) розробили гібридну модель Transformer-Hawkes для детектування інформаційних операцій шляхом аналізу реакцій соціальних мереж у часі [6]. Основний результат полягає в покращенні точності виявлення прихованої координації агентів. Luceri et al. (2023) запропонували архітектуру на основі графових нейронних мереж, що забезпечує високу ефективність у виявленні скоординованої неавтентичної активності у соціальних платформах [7]. Результати засвідчили точність вище 95% на реальних датасетах. Zeng et al. (2023) представили модель SIASM, яка поєднує структурну ентропію та моделювання соціальних ботів для вивчення прихованих тактик поширення дезінформації [8]. Дослідження акцентує на ролі складних топологій мереж у забезпеченні ефективності операцій. Pastor-Galindo et al. (2025) на платформі arXiv представили типологію інформаційних операцій (FIMI) з прикладами масових кампаній, що охоплюють понад 40 країн [9]. Автори акцентують на необхідності багаторівневих систем виявлення та розпізнавання складових впливу. Menn (2025) описує кейс цілеспрямованого інфікування генеративних моделей фейковими твердженнями, що потрапляють до відповідей чат-ботів. Стаття ілюструє новий вектор атак – LLM-grooming – та обґрунтовує потребу у контролі достовірності тренувальних даних [10].

Цей аналіз поглиблює розуміння актуальних підходів до моделювання ІО, їх інтеграції з сучасними технологіями та підкріплює вибір методів, запропонованих у цій роботі (таб. 1.).

Таблиця 1.

Загальна оцінка проаналізованих підходів

Напрям	Основні тренди	Виклики
ABM + LLM	Автономна поведінка агентів з генеративними властивостями	Високе навантаження, етичні питання

Напрям	Основні тренди	Виклики
Еволюційні алгоритми	Адаптація стратегій у динамічних мережах	Оптимізаційні витрати
Графові ML-підходи	Висока точність виявлення ІО	Поведінкові зміни мереж
Епідеміологічні моделі	Адекватне врахування часу та втручання	Ліміти щодо реалістичності
Алгоритмічна радикалізація	Виявлено ефект filter bubbles	Прозорість систем

Аналіз сучасних досліджень засвідчує, що методи моделювання інформаційних операцій швидко еволюціонують у бік комплексних багаторівневих систем, які поєднують симуляційні, мережеві та машинно-навчальні підходи. Зростаюча роль штучного інтелекту, мовних моделей та адаптивної динаміки мереж зумовлює необхідність створення нових агентно-орієнтованих архітектур. Саме ці виклики та тенденції стали підґрунтям для розробки моделі, запропонованої у цій статті, яка інтегрує ключові алгоритмічні блоки для вивчення складної взаємодії в інформаційному середовищі соціальних мереж.

Викладення основного матеріалу.

За [11-13] під поняттям інформаційна операція слід розуміти комплекс цілеспрямованих дій, що здійснюються в інформаційному середовищі з метою впливу на інформаційні ресурси, інформаційні процеси, інфраструктуру або поведінку цільової аудиторії (суспільства, держави, організації) з метою досягнення політичних, військових або соціальних цілей.

У контексті інформаційної безпеки будь якої держави інформаційні операції розглядається як інструмент стратегічного впливу, що поєднує технічні, психологічні, соціальні та когнітивні механізми.

Інформаційні операції є багаторівневими, тобто мають тактичний, операційний, стратегічний рівні, при цьому на кожному із зазначених рівнів наявні певні цілі (вплив, дестабілізація, маніпуляція) та об’єкти/цільові аудиторії. Їм властива тривалість у часі та взаємодія з іншими кампаніями (військовими, політичними), а також використання інтегрованих засобів (соціальні мережі, ЗМІ, боти, вразливості ІТ-систем).

Таблиця 2.

Порівняння з близькими поняттями

Поняття	Коротке визначення	Основна відмінність від ІО
Інформаційна атака	Окрема цілеспрямована дія або серія дій, що має на меті завдати шкоди інформаційній інфраструктурі, даним або когнітивному стану аудиторії	Є *елементом* або *тактичним прийомом* у межах ширшої ІО. Має вузьку спрямованість, зазвичай разову.
Інформаційна акція	Одноразовий захід, спрямований на передавання певного повідомлення чи вплив на публіку (наприклад, флешмоб, гасло, кампанія в Twitter)	Обмежена у часі, не є багатокомпонентною. Виступає *одиницею тактики* у межах ІО або кампанії.
Інформаційна кампанія	Організований комплекс дій з поширення повідомлень, що триває певний період і спрямований на формування образу, ставлення, поведінки	Орієнтована здебільшого на **легітимну публічну комунікацію** (наприклад, передвиборча кампанія, просвітницька діяльність). ІО може включати нелегітимні засоби, маніпуляцію, дезінформацію.
Інформаційна операція	Системна стратегія впливу на інформаційне середовище для досягнення політичних, військових або гібридних цілей	Є **інтегрованим стратегічним поняттям**, що включає атаки, акції, кампанії в єдину систему впливу.

Як демонструє аналіз останніх досліджень основним середовищем для здійснення інформаційних операцій на сьогодні стали соціальні мережі кіберпростору завдяки мультимедійності, інтерактивності, можливості зворотного зв'язку, горизонтальності зв'язків, відсутності посередників, позагеографічності або позапросторовості.

Такі властивості дозволяють організувати процес за таким алгоритмом: ідентифікація ключових вузлів мережі (інфлюенсерів), які мають стратегічне значення для поширення контенту; застосування методу Монте-Карло як симуляції стохастичного прогнозування поведінки агентів у рамках заданої мережевої структури; оптимізація стратегій агентів, що може включати еволюційні алгоритми або генетичні моделі для адаптації до змін у середовищі; виявлення спільнот, що дозволяє визначити ключові групи взаємодії.

На основі змін у поведінці агентів система приймає рішення щодо оновлення параметрів або залишення моделей без змін. Таким чином формується циклічна адаптивна взаємодія між агентами, соціальною мережею та процесами впливу. На виході – оновлений інформаційний стан мережі, що включає реакції, впливи, поширення та адаптацію до зворотного зв'язку.

Графічно цей алгоритм можна представити як загальну архітектуру інформаційної операції, реалізованої в соціальній мережі.

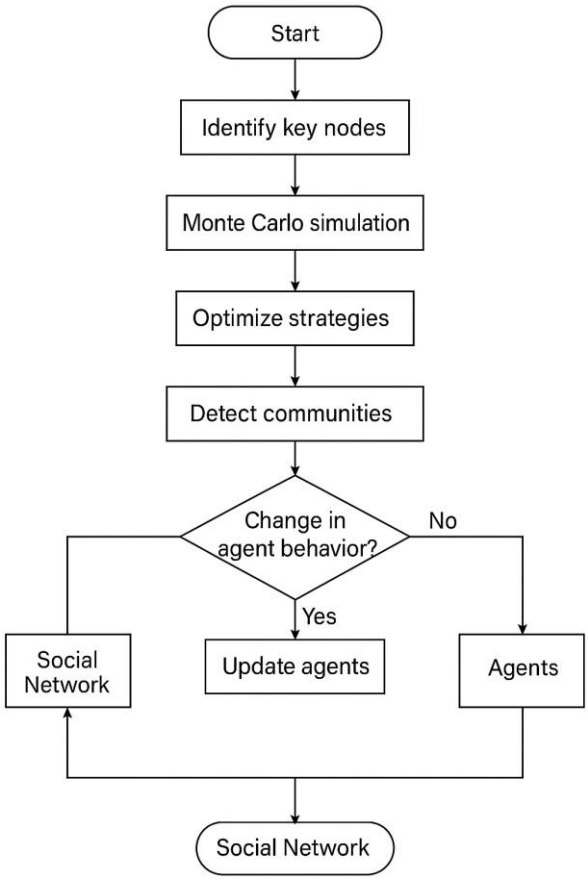


Рис. 1. Архітектура та логіка функціонування інформаційної операції в соціальній мережі.

Процес моделювання інформаційної операції в соціальній мережі розпочинається з етапу ідентифікації ключових агентів – так званих лідерів думок (інфлюенсерів), які є стратегічно важливими вузлами та мають високий потенціал впливу на мережу. Наступним кроком є ймовірнісне прийняття рішень цими агентами, що відбувається на основі оцінки стану середовища та інформаційного впливу.

Після цього виконується виявлення спільнот (community detection), що дозволяє визначити кластери агентів, між якими поширюється інформація. На кожному кроці існує можливість повернення до попереднього етапу, якщо мережа ще не стабілізувалась.

Паралельно процесу функціонує підпроцес на рівні окремого агента, поданий як скінченний автомат. Агент може перебувати в станах: uninfluenced (не зазнав впливу), influenced (під впливом), old (вийшов із зони актуального впливу). Перехід між станами моделюється з урахуванням зворотного зв'язку. Якщо агент підпадає під інформаційний вплив, він переходить у стан influenced, а згодом – у старий (old), або повертається до початкового стану залежно від мережевого контексту.

На завершальному етапі моделювання перевіряється, чи досягнуто кінцевої мети (наприклад, стабільності мережі або повного поширення інформації). Якщо ні, процес повертається до попередніх фаз для подальшого циклу моделювання.

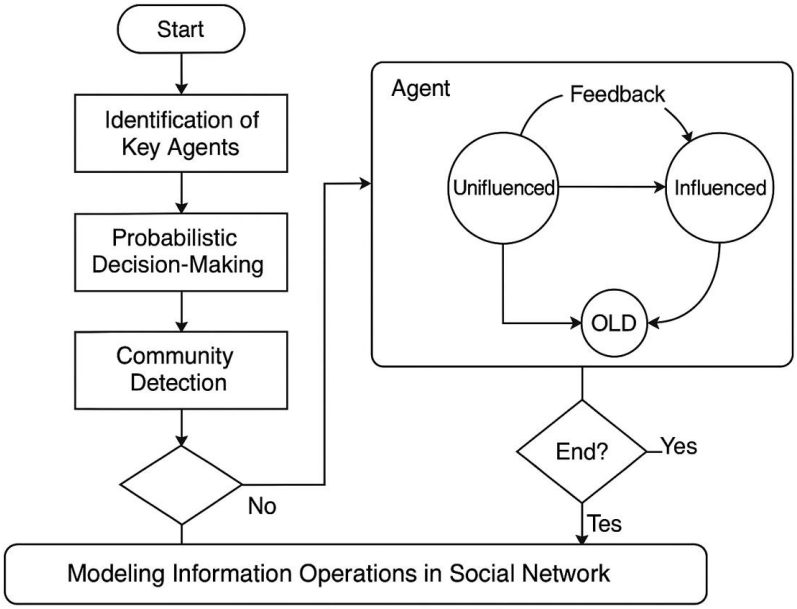


Рис. 2. Схема моделювання інформаційної операції в соціальній мережі із зазначенням логіки агентів.

Під час використання агентного моделювання для моделювання інформаційних операцій у соціальних мережах можна застосувати різноманітні методи та алгоритми, які забезпечують реалізацію цих моделей. Нижче проаналізовано основні з них.

Алгоритми на основі теорії ігор.

- Метод «Наївного байєсівського гравця» використовується для моделювання поведінки агентів в умовах невизначеності, коли вони ухвалюють рішення на основі ймовірнісних оцінок.

Так, агент A_i оцінює ймовірність стану середовища S як P(S|O_i), де O_i — спостереження. Рішення приймається згідно з: argmax_a ∑_S P(S|O_i)·U(a, S)

- Метод динамічних ігор дає змогу моделювати стратегії агентів, які взаємодіють із плином часу, беручи до уваги як минулу, так і майбутню поведінку інших агентів.

Формалізуються як послідовність стратегічних взаємодій G = (N, A_i, T, U_i(t)), де агенти оптимізують стратегію з урахуванням змін у середовищі.

Методи поширення впливу.

- Алгоритм поширення (максимізації) впливу (Influence Maximization) застосовується для визначення ключових агентів, які можуть максимізувати поширення інформації в мережі.

Для заданої мережі G = (V, E) знаходять підмножину S ⊆ V, що максимізує охоплення: max_{|S| ≤ k} σ(S), де σ(S) – очікувана кількість активованих вузлів.

- Алгоритм відсікання та обрізки (Cutoff and Pruning) використовується для скорочення розміру мережі, зберігаючи при цьому найбільш значущі зв'язки для поширення інформації. Видаляються вузли та ребра з низьким коефіцієнтом центральності, зберігаючи функціональне ядро мережі.

Алгоритми машинного навчання.

- Методи класифікації (наприклад, SVM, Decision Trees) застосовуються для класифікації агентів за групами залежно від їхньої поведінки або сприйняття інформації.

SVM або Decision Tree – модель f: X → Y, де X – ознаки агента, Y – мітка класу (інфлюенсер, ретранслятор тощо).

- Методи кластеризації (наприклад, K-Means, DBSCAN) використовуються для виявлення спільнот або груп агентів зі схожими характеристиками.

K-Means або DBSCAN – розбиття простору на кластери $C = \{C_1, C_2, ..., C_k\}$ на основі евклідових відстаней.

Еволюційні алгоритми.

- Генетичні алгоритми застосовуються для оптимізації стратегій агентів, даючи їм змогу «еволюціонувати» та адаптуватися до змін в інформаційному середовищі.

Стратегії кодуються як хромосоми, над якими виконуються відбір, кросовер, мутація. Оптимізація функції U_i = f(strategy_i, network_context).

- Алгоритм диференціальної еволюції використовується для пошуку оптимальних параметрів моделі, враховуючи адаптивні здібності агентів.

Пошук оптимальних параметрів: x_{i}^{t+1} = x_r^1 + F·(x_r^2 - x_r^3)

Методи мережевого аналізу.

- Алгоритми пошуку шляхів (наприклад, Dijkstra (Дéйкстра) Алгоритм): використовуються для визначення найкоротших шляхів поширення інформації через мережу.

Визначення найкоротших шляхів: d(u, v) = min ∑_{e ∈ path(u,v)} w(e)

- Алгоритми виявлення спільнот (наприклад, Girvan-Newman, Louvain) дають змогу ідентифікувати групи агентів, які активно взаємодіють один з одним, що важливо для моделювання кластерів інформації. Виділення спільнот шляхом максимізації модульності Q.

Алгоритми прийняття рішень.

- Метод Монте-Карло (MCMC) використовується для моделювання ухвалення рішень агентами, що ґрунтуються на ймовірнісних оцінках майбутніх станів.

- Алгоритми рендерингу рішень на основі корисності застосовуються для моделювання раціональних агентів, які приймають рішення, максимізуючи свою корисність.

Методи обробки великих даних.

- MapReduce застосовується для опрацювання та аналізу великих обсягів даних про соціальні мережі, даючи змогу масштабувати моделювання на великі мережі.

- Алгоритми розподіленого обчислення використовуються для паралельного опрацювання даних і виконання моделювання на кількох вузлах.

Методи динамічного моделювання.

- Алгоритми зворотного зв'язку (Feedback Loops) використовуються для моделювання динамічних процесів у соціальних мережах, таких як зворотні зв'язки в поширенні інформації.

Циклічні зміни в графі станів: s_{t+1} = f(s_t, I_t)

- Метод скінченних автоматів: застосовується для моделювання станів агентів та їхніх переходів між цими станами.

Агенти переходять між станами згідно з правилами $\delta(s, a) \rightarrow s'$

Кожен із цих методів і алгоритмів може бути адаптований і використаний залежно від специфіки завдань, пов'язаних із моделюванням інформаційних операцій у

соціальних мережах. Їх вибір і комбінація залежать від особливостей моделі та цілей дослідження.

Для агентного моделювання інформаційних операцій у соціальних мережах можна створити комбінацію методів, яка дасть змогу моделювати складні соціальні процеси, розповсюдження інформації та взаємодію агентів. Автором запропоновано наступні комбінації методів, яка поєднують в собі різні аспекти агентної поведінки та динаміки соціальних мереж:

1) Алгоритми поширення впливу + Алгоритми ухвалення рішень.

- Алгоритм поширення впливу (Influence Maximization): цей метод застосовується для визначення ключових агентів (інфлюенсерів) у соціальній мережі, які здатні максимізувати поширення інформації. За допомогою цього алгоритму ідентифікуються вузли, які можуть ефективно впливати на більшу частину мережі.

- Метод Монте-Карло (MCMC). Після визначення інфлюенсерів раціонально застосовувати метод Монте-Карло для моделювання рішень агентів, ухвалених на основі ймовірнісних оцінок. Це дасть змогу оцінити, як інформація поширюватиметься через мережу з урахуванням непередбачуваності поведінки окремих агентів.

2) Еволюційні алгоритми + Методи мережевого аналізу.

- Генетичні алгоритми: застосування цього методу підходить для оптимізації стратегій агентів, які беруть участь в інформаційних операціях. Він дасть змогу агентам адаптуватися та покращувати свої стратегії з плином часу, що особливо важливо для динамічних соціальних мереж.

- Алгоритм виявлення спільнот (Louvain) для виявлення спільнот всередині соціальної мережі допоможе краще зрозуміти, як інформація поширюється в рамках кластерів і як ефективно націлюватися на ключові групи агентів.

3) Методи кластеризації + Методи динамічного моделювання.

- Кластеризація методом K-Means застосовується для сегментації агентів на групи зі схожими характеристиками або поведінкою. Цей метод корисний для розуміння того, як різні типи агентів можуть реагувати на інформаційні операції.

- Алгоритми зворотного зв'язку (Feedback Loops) слід використати для моделювання динамічних процесів у мережі, таких як зміна думок або адаптація агентів у відповідь на зміни в інформаційному середовищі. Цей метод дає змогу враховувати, як минула інформація впливає на майбутню поведінку агентів.

4) Методи обробки великих даних + Агентне моделювання.

- MapReduce: використання MapReduce для опрацювання великих обсягів даних про взаємодію агентів та їхню активність у соціальних мережах дасть змогу масштабувати моделювання на великі мережі та враховувати величезну кількість даних, що важливо для реалістичних симуляцій.

- Агентне моделювання. Всі перераховані вище методи будуть інтегровані в загальний агентний фреймворк, який дасть змогу моделювати поведінку агентів на мікрорівні та їхню взаємодію на макрорівні, що призведе до створення більш точних і достовірних симуляцій інформаційних операцій.

Ця комбінація методів забезпечує всебічний підхід до агентного моделювання, враховуючи як стратегічну поведінку агентів, так і динамічну взаємодію в соціальних мережах. Вона дає змогу моделювати складні сценарії, як-от поширення дезінформації, вплив лідерів думок, динаміка зміни думок у мережі та адаптація агентів у реальному часі.

Для побудови агентної моделі, спрямованої на моделювання інформаційних операцій у соціальних мережах, можна запропонувати також поєднання одразу п’яти методів у визначеній послідовності з урахуванням їхньої інтеграції в загальну агентну модель. Кожен алгоритм відповідатиме за певний аспект моделювання, і разом вони забезпечать всебічний і комплексний аналіз інформаційних операцій у соціальних

мережах. Нижче запропоновано комбінацію із зазначенням мети та обґрунтування кожного методу, а також його ролі в моделюванні.

1. Алгоритм поширення впливу (Influence Maximization)

Мета: визначення ключових агентів (інфлюенсерів) у соціальній мережі, здатних максимізувати поширення інформації.

Обґрунтування: Інформаційні операції часто залежать від ключових фігур, які можуть значно вплинути на всю мережу. Цей алгоритм дає змогу визначити, які агенти можуть максимально ефективно поширювати інформацію.

Роль: на першому етапі модель визначає ключові вузли в соціальній мережі, які мають найбільший потенціал для поширення інформації. Ці вузли будуть пріоритетними агентами, через яких розпочнеться інформаційна операція.

2. Метод Монте-Карло (MCMC)

Мета: моделювання ухвалення рішень агентами, що ґрунтуються на ймовірнісних оцінках і впливі оточуючих.

Обґрунтування: у реальних соціальних мережах поведінка агентів часто невизначена і може залежати від безлічі факторів. Метод Монте-Карло дає змогу змоделювати цю невизначеність і оцінити, як інформація поширюватиметься за різних сценаріїв.

Роль: після того як ключові вузли визначено, метод Монте-Карло моделює ймовірнісні рішення агентів у мережі. Це дає змогу оцінити, як інформація поширюватиметься, враховуючи випадкові фактори та невизначеність у поведінці агентів.

3. Генетичні алгоритми.

Мета: оптимізація стратегій агентів у процесі інформаційних операцій.

Обґрунтування: інформаційні операції динамічні, і агенти повинні адаптуватися до змін у мережі. Генетичні алгоритми дають змогу агентам «еволюціонувати», покращуючи свої стратегії та адаптуючись до нових умов.

Роль: на етапі оптимізації модель використовує генетичні алгоритми для поліпшення стратегій агентів. Це може включати адаптацію до змін у мережі або розробку нових методів поширення інформації на основі реакції агентів.

4. Алгоритм виявлення спільнот (Louvain).

Мета: ідентифікація спільнот і кластерів усередині соціальної мережі для більш точного моделювання поширення інформації.

Обґрунтування: інформація в соціальних мережах часто поширюється через кластери або спільноти. Алгоритм Louvain допомагає виявити ці кластери, що дає змогу моделювати поширення інформації з урахуванням структури мережі.

5. Алгоритми зворотного зв'язку (Feedback Loops).

Мета: моделювання динамічних змін у поведінці агентів і поширенні інформації.

Обґрунтування: інформаційні операції мають циклічний характер, де кожне нове повідомлення або дія може впливати на подальшу поведінку агентів. Зворотні зв'язки дають змогу враховувати ці зміни, роблячи модель реалістичнішою.

Роль: зворотний зв'язок дає змогу моделі враховувати динамічні зміни, як-от зміна думок агентів або адаптація їхньої поведінки у відповідь на отриману інформацію. Цей компонент робить модель реалістичнішою і дає змогу симулювати ефекти, які проявляються з плином часу.

Приклад сценарію:

Початковий етап. Модель визначає ключових агентів з використанням алгоритму поширення впливу. Це задає стартові умови для інформаційної операції.

Поширення інформації. Метод Монте-Карло моделює, як інформація поширюватиметься через мережу, враховуючи ймовірності та випадкові ефекти. Одночасно, алгоритм Louvain допомагає зрозуміти, як інформація поширюється всередині та між спільнотами.

Адаптація та оптимізація. У процесі поширення інформації генетичні алгоритми оптимізують стратегії агентів, адаптуючи їх до нових умов. Наприклад, агенти можуть змінювати свої стратегії на основі реакції інших агентів.

Зворотний зв'язок і динаміка. Алгоритми зворотного зв'язку враховують зміни, що відбуваються в часі, як-от зміна громадської думки або вплив нових інформаційних подій.

Застосування всіх п'яти алгоритмів одночасно дає змогу створити гнучку, адаптивну та динамічну модель, яка здатна імітувати складні процеси інформаційних операцій у соціальних мережах. Ці алгоритми, працюючи разом, дають змогу враховувати різноманітні аспекти поведінки агентів, структуру мережі, імовірнісні ефекти та динамічні зміни, що робить модель реалістичнішою та потужнішою.

Обґрунтування вибору. Ця комбінація методів і алгоритмів дає змогу створити модель, яка враховує як мікродинаміку поведінки окремих агентів, так і макродинаміку мережі загалом. Алгоритми поширення впливу та виявлення спільнот допомагають зрозуміти, як інформація поширюється і через які вузли. Методи Монте-Карло і генетичні алгоритми забезпечують гнучкість і адаптивність моделі, даючи їй змогу ефективно моделювати невизначеність і зміни в стратегіях агентів. Алгоритми зворотного зв'язку додають важливий аспект динаміки, даючи змогу моделі враховувати циклічні зміни та адаптацію агентів у реальному часі.

Ця комбінація робить модель комплексною і придатною для аналізу широкого спектра сценаріїв інформаційних операцій, включно з поширенням дезінформації, впливом лідерів думок, а також динамічною зміною думок і взаємодій усередині соціальної мережі.

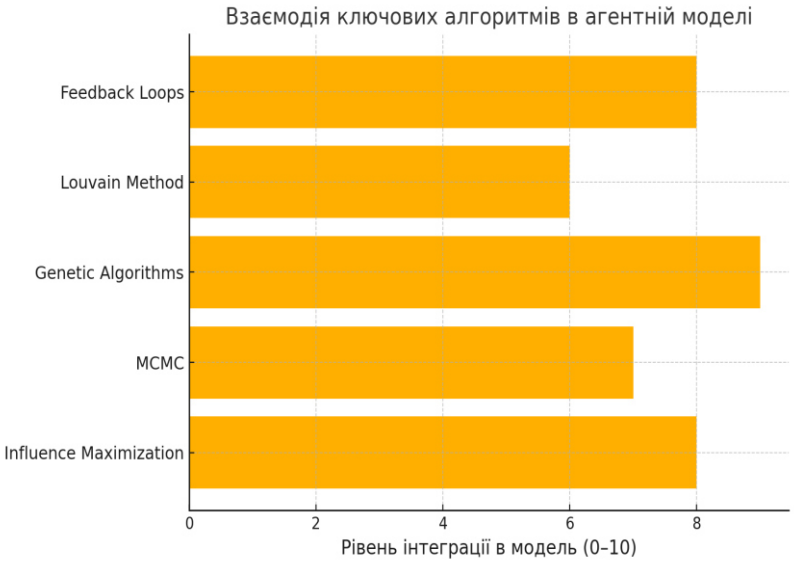


Рис. 3. Взаємодія алгоритмів в межах агентної моделі для моделювання інформаційних операцій у соціальних мережах.

Висновки. У статті проведено систематизацію сучасних методів та алгоритмів, що використовуються для моделювання інформаційних операцій у соціальних мережах. Показано, що поєднання агентного моделювання з такими підходами, як алгоритми поширення впливу, еволюційна оптимізація, теорія ігор, методи кластеризації, машинне навчання та мережевий аналіз, дає змогу створити комплексні симуляційні моделі, здатні точно імітувати динаміку інформаційного простору.

Застосування цих комбінацій дозволяє моделювати не лише розповсюдження інформації, а й стратегічну взаємодію між агентами, зміну їхніх стратегій, вплив структури мережі на ефективність операцій та адаптацію системи до нових загроз. Підкреслено важливість включення зворотного зв’язку та використання алгоритмів

обробки великих даних для масштабованості моделей. Отримані результати мають практичне значення для розробки систем виявлення та протидії інформаційним загрозам, а також для стратегічного планування у сфері інформаційної безпеки.

Список літератури

1. Adriana Iamnitchi, Lawrence O. Hall, Sameera Horawalavithana, Frederick Mubang, Kin Wai Ng, John Skvoretz Modeling information diffusion in social media: data-driven observations, Frontiers in Big Data, Cambridge Review (2024) May 2023, DOI:10.3389/fdata.2023.1135191
2. Islam, M. S., et al. (2024). Information diffusion: analysis, process, model, deployment and application. *Knowledge Engineering Review*. https://www.cambridge.org/core/journals/knowledge-engineering-review/article/information-diffusion-analysis-process-model-deployment-and-application/1B4FB3A4C369592177CB4678B739E208
3. Liao, Y., et al. (2024). Epidemic-like modeling of misinformation spread with sentiment and intervention timing. *Scientific Reports*. https://www.nature.com/articles/s41598-024-69657-0
4. Chen Gao, Xiaochong Lan, Nian Li, Yuan Yuan, Jingtao Ding, Zhilun Zhou, Fengli Xu & Yong Li Large language models empowered agent-based modeling and simulation: a survey and perspectives, Humanities and Social Sciences Communications volume 11, Article number: 1259 (2024) https://www.nature.com/articles/s41599-024-03611-3?utm_source=chatgpt.com
5. Ding, Y., et al. (2024). Agent-based modeling meets generative AI in social network simulations. *ResearchGate*. https://www.researchgate.net/publication/386112826
6. Quyu Kong, Pio Calderon, Rohit Ram, Olga Boichak Interval-censored Transformer Hawkes: Detecting Information Operations using the Reaction of Social Systems DOI:10.48550/arXiv.2211.14114 https://www.researchgate.net/publication/365784012_Interval-censored_Transformer_Hawkes_Detecting_Information_Operations_using_the_Reaction of_Social_Systems
7. Luca Luceri, Valeria Pantè, Keith Burghardt, Emilio Ferrara Unmasking the Web of Deceit: Uncovering Coordinated Activity to Expose Information Operations on Twitter, The Web Conference 2024
8. Zeng, J., et al. (2023). SIASM: Structural entropy-driven model for social bot infiltration. *arXiv preprint*. https://arxiv.org/abs/2312.08098
9. Pastor-Galindo, J., et al. (2025). Influence operations in social networks: FIMI taxonomy and detection. *arXiv preprint*. https://arxiv.org/abs/2502.11827
10. Menn, J. (2025). Russia seeds chatbots with lies. *The Washington Post*. https://www.washingtonpost.com/technology/2025/04/17/llm-poisoning-grooming-chatbots-russia/
11. Rid, T. (2020). *Active Measures: The Secret History of Disinformation and Political Warfare*. Farrar, Straus and Giroux.
12. NATO (2017). *NATO Terminology Database – Information Operations*.
13. Бєлєсков, М. (2021). *Інформаційна війна у гібридному конфлікті: підходи та методи*. Центр Разумкова.

О.О. Васильєва

ALGORITHMS AND METHODS FOR MODELING INFORMATION OPERATIONS IN SOCIAL NETWORKS

O.O. Vasylieva

Chernihiv Polytechnic National University
95, Shevchenko street, Chernihiv, 14030, Ukraine
Email:olga.vasiljeva37@gmail.com

The article analyzes latest scientific experiments on modeling information operations in social networks in cyberspace, which deepens the understanding of the relevance of approaches and their integration with modern technologies. A general assessment of the analyzed approaches is provided, and attention is focused on the growing role of artificial intelligence, language models, and adaptive network dynamics, which necessitates the creation of new agent-oriented architectures. The term “information operations” is presented and compared with related concepts. Focusing on the main characteristics of social networks – multimedia, interactivity, feedback capabilities, horizontal connections, absence of intermediaries, and non-geographical or non-spatial nature – the architecture and logic of information operations in social networks are proposed. A general algorithm for information operations in social networks of cyberspace is presented graphically, including the identification of key agents, decision-making by these agents, and the identification of communities and clusters of agents. Classes of methods and algorithms used to model information operations in social networks are considered. Combinations of methods are proposed from the point of view of agent-based modeling, combining Influence propagation algorithms + Decision-making algorithms; Evolutionary algorithms + Network analysis methods; Clustering methods + Dynamic modeling methods; Big data processing methods + Agent-based modeling. To build an agent model aimed at modeling information operations in social networks, a combination of five methods in a specific sequence is proposed, taking into account their integration into the general agent model, and an example of a scenario for comprehensive modeling of information operations in social networks in cyberspace is given.
Keywords: information operations, social networks, agent-based modeling, evolutionary algorithms, game theory, machine learning, MCMC, Louvain, Influence Maximization.

DOI 10.15276/imms.v15.no2.187
УДК 004.56

Informatics and Mathematical Methods in Simulation
Vol.15 (2025), No.2, pp. 187-195

РОЗРОБКА ВДОСКОНАЛЕНОЇ МЕТОДИКИ ОПОВІЩЕННЯ НАСЕЛЕННЯ ПРО КІБЕРІНЦИДЕНТИ

О.І. Гарасимчук, Т.А. Костишин, О.О. Любчик, М.Є. Швед

Національний університет «Львівська політехніка»
12, Степана Бандери вул., м. Львів, 79013, Україна
Emails: oleh.i.harasymchuk@lpnu.ua, tetiana.kostyshyn.kb.2021@lpnu.ua, olha.liubchyk.kb.2021@lpnu.ua, mariia.y.shved@lpnu.ua

В умовах активного впровадження цифрових технологій у суспільне життя та як наслідок зростання кількості кіберзагроз, особливо в контексті повномасштабної війни в Україні, питання ефективного інформування населення про кіберінциденти набуває критичної важливості. Традиційні засоби масової інформації не завжди здатні забезпечити необхідну швидкість та охоплення для оперативного оповіщення, що перетворює соціальні мережі та месенджери на ключові канали комунікації в сучасних умовах. У даній статті розглянуто наявні системи оповіщення населення про кіберінциденти та на основі аналізу запропоновано покращену методику донесення інформації до суспільства. Головною метою цього дослідження є розробка вдосконаленої системи оповіщення населення про кіберінциденти із використанням соціальних мереж, які на даний момент є найефективнішою платформою для швидкого поширення інформації. Дослідження доповнює наявні моделі комунікації в умовах кіберзагроз новими елементами цифрової взаємодії та підвищенням участі користувачів. Запропоновані рекомендації можуть бути застосовані державними структурами для удосконалення механізмів оповіщення за рахунок використання покращеної структури повідомлення, чат-ботів Telegram, візуалізації, інтерактивного навчання, додаткових каналів зворотного зв'язку та залучення інфлюенсерів. Подане у статті дослідження має якісний характер і базується на порівняльному аналізі інформаційних повідомлень у соціальних мережах від державних установ. У результаті дослідницької роботи запропоновано покращену модель інформування про кіберінциденти з використанням комплексного, інноваційного підходу, адаптованого до українських умов. У подальшій перспективі доцільно оцінити ефективність нової моделі після її впровадження на основі порівняння досягнутих результатів із наявними на даний момент.

Ключові слова: кіберінцидент, соціальні мережі, оповіщення, кібербезпека, комунікація.

Вступ. Зі стрімкою цифровізацією світу кібербезпека стає одним із провідних напрямків, що потребують розвитку та інвестицій, адже інформаційні технології, які тепер доступні суспільству впливають як і на кожного громадянина окремо так і на державу загалом. Разом із цим зростає й кількість кіберінцидентів – атак, шахрайств, витоків даних, що несуть загрозу як окремим користувачам, так і національній безпеці в цілому [1]. Особливо гостро ця проблема стоїть для України, що активно інтегрується у глобальний цифровий простір і водночас стикається із численними кіберзагрозами, у тому числі на фоні збройного конфлікту [2-4].

Розвитку потребують багато напрямків кібербезпеки, але один із головних для забезпечення захисту громадян державою – це вчасне і ефективне інформування населення про виникнення кіберінцидентів. Звичайні користувачі повинні володіти інформацією про загрози та дієві способи їм протидії щоб значно знизити ризики та мінімізувати негативний вплив на суспільство. Проте, часто у кризових ситуаціях ми стикаємось із реальністю, в якій традиційні засоби масової інформації такі як телебачення та інформаційні видання не можуть забезпечити необхідну швидкість оповіщення і охоплення.

Соціальні мережі та месенджери, завдяки своїй масовості, оперативності та інтерактивності, стають основними каналами комунікації для інформування громадськості у цифрову епоху. Вони дають змогу не лише швидко поширювати важливу інформацію, а й отримувати зворотний зв'язок від користувачів, що сприяє формуванню більш активної та свідомої аудиторії. Проте використання цих платформ для поширення повідомлень про кіберінциденти пов'язане з низкою викликів — від контролю якості інформації та протидії дезінформації до забезпечення конфіденційності та захисту персональних даних.

Мета дослідження. Метою даного дослідження є проаналізувати особливості подання інформації про кіберінциденти в різних соціальних мережах, що регулярно висвітлюють кіберзагрози та запропонувати покращений метод оповіщення.

Теоретичні основи дослідження. Телеграм-канал Держспецзв'язку став одним з основних джерел інформації про кібератаки на державні органи та критичну інфраструктуру України. Подача інформації тут характеризується оперативністю, офіційністю та технічною конкретикою. Як правило, повідомлення містять: (1) короткий виклад суті інциденту, (2) технічні деталі або ідентифікатори загрози, (3) посилання на додаткову інформацію (сайт CERT-UA чи офіційний веб-сайт Служби) та (4) рекомендації або заклик до дій [5].

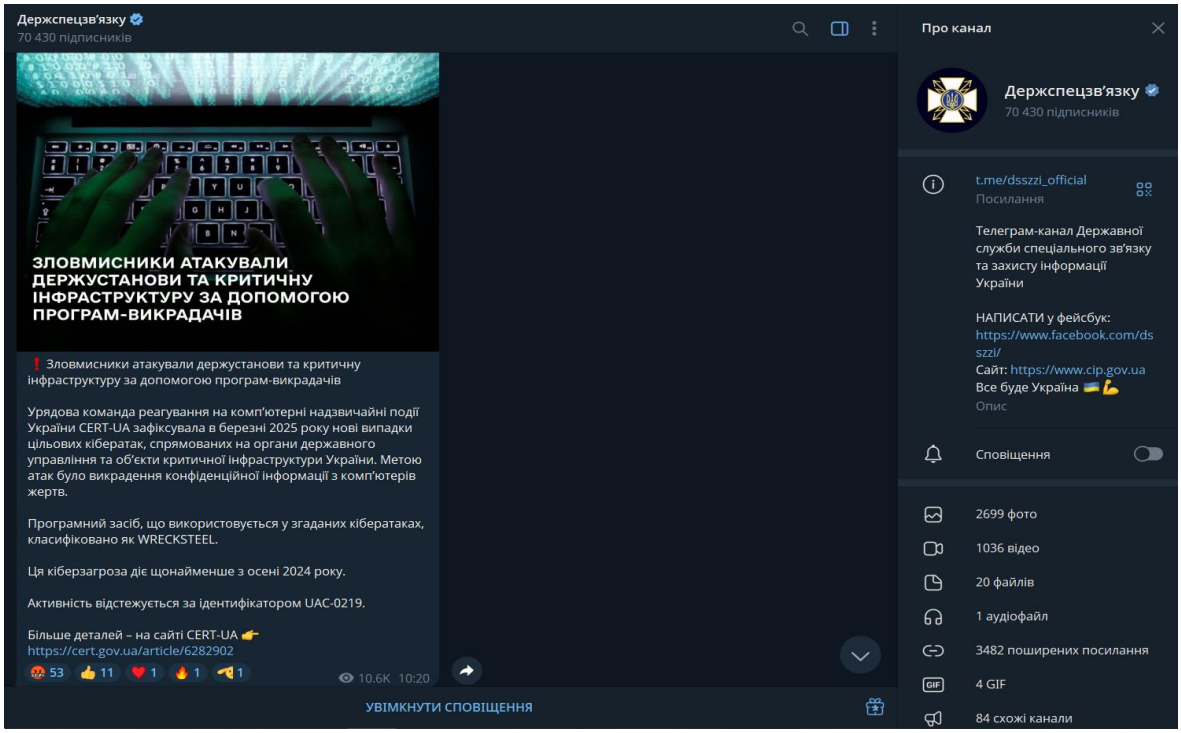


Рис. 1. Telegram-канал Держспецзв'язку

Важливим аспектом є наявність практичних порад у повідомленнях Держспецзв'язку. Часто після опису кібератаки публікуються рекомендації фахівців, як запобігти подібним атакам або зменшити їхній вплив. Канал орієнтований одразу на кілька категорій аудиторії: і на технічних спеціалістів (яким потрібні детальні кроки для захисту), і на широке коло користувачів, які можуть постраждати (їм пояснюють, куди повідомити про інцидент). Для ширшої публіки Держспецзв'язку також публікує загальні поради з кібергігієни у зручному форматі. Наприклад, окремі дописи присвячені простим правилам, як убезпечити дані на зовнішніх носіях (флешках, дисках): сканувати антивірусом, вимкнути автозапуск, не використовувати чужі носії тощо. Такі матеріали подаються списком з підкресленням ключових дій, що зручно для сприйняття.

Іншим ключовим джерелом є комунікації Кіберполіції України через Telegram, Facebook та X. Подача інформації від кіберполіції має свою специфіку: акцент на захисті громадян

від онлайн-злочинів (шахрайств, фішингу, крадіжок даних) та на результатах розслідувань кіберзлочинів. Стиль повідомлень більш прикладний, з роз'ясненнями доступною мовою, без надмірно технічних деталей – адже орієнтований на масову аудиторію користувачів Інтернету.

Під час війни кіберполіція фіксує численні випадки шахрайства, що експлуатують тему війни та волонтерства: фейкові збори коштів "на ЗСУ", фальшиві виплати постраждалим чи продаж неіснуючих товарів для військових [6]. Як тільки з'являється нова схема, правоохоронці готують попередження у соціальних мережах. Структура таких попереджень типово містить: (1) застереження-заклик до громадян бути пильними, (2) опис того, як працює схема шахраїв, (3) поради, як не стати жертвою.

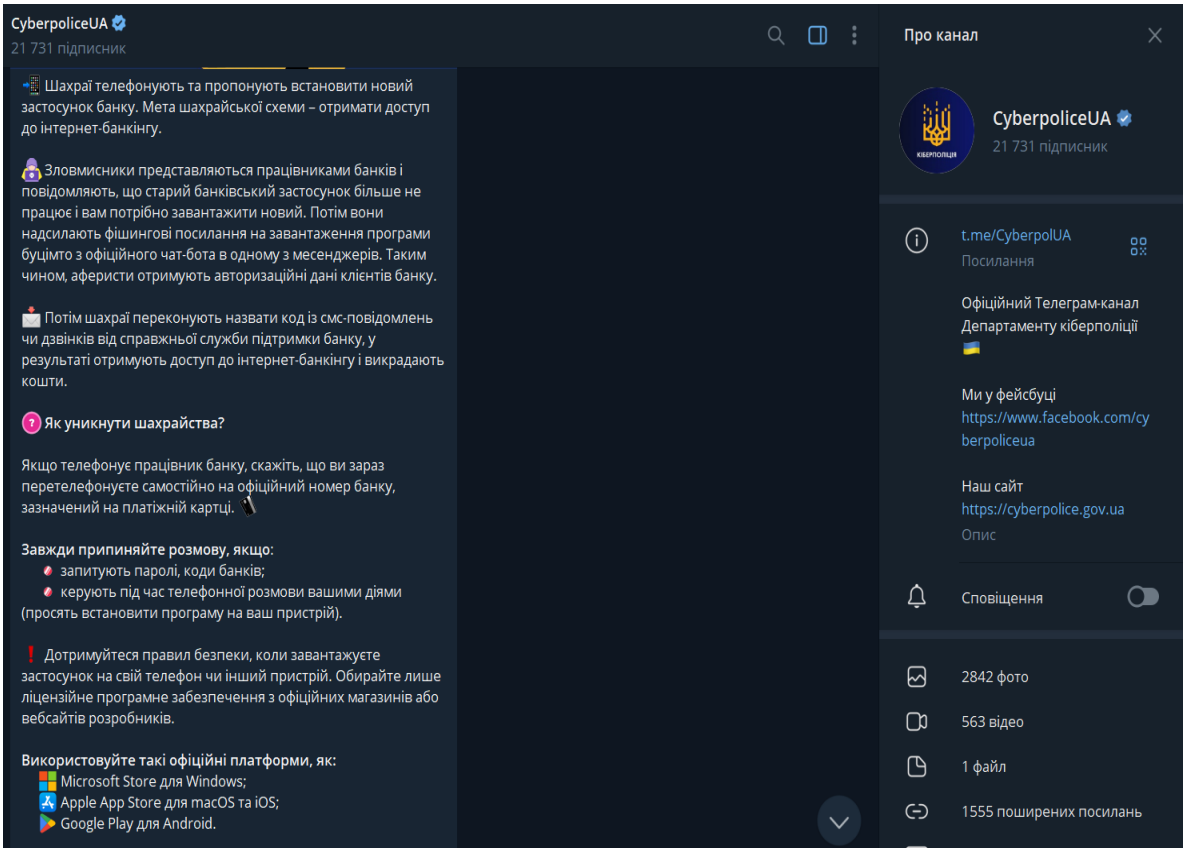


Рис. 2. Telegram-канал кіберполіції CyberpoliceUA

Кіберполіція зосереджується на безпеці простих користувачів, дає практичні поради і негайно реагує на нові шахрайські схеми, що виникають у зв'язку з війною. Завдяки соцмережам ці попередження швидко ширяться – їх підхоплюють регіональні адміністрації, ЗМІ, освітні установи, дублюючи на своїх ресурсах [7], що множить охоплення.

Варто згадати, кіберполіція також активно використовує Facebook та X для кіберпопереджень. Ці матеріали доповнюють Telegram-попередження, роблячи їх більш доступними широкій аудиторії. За рахунок поєднання різних соцмереж кіберполіція охоплює різні вікові та соціальні групи – від школярів до пенсіонерів[8-9] – і доносить до них знання про актуальні кіберзагрози воєнного часу. До прикладу, сторінка кіберполіції могла дублювати повідомлення про нові кібератаки, а користувачі Facebook та X активно ними ділилися. Однак, месенджери на кшталт Telegram все ж перевершили Facebook та X за швидкістю охоплення [10]: у Telegram люди підписані на канали і отримують миттєві сповіщення, тоді як у Facebook та X алгоритми стрічки могли показати важливе попередження із затримкою.

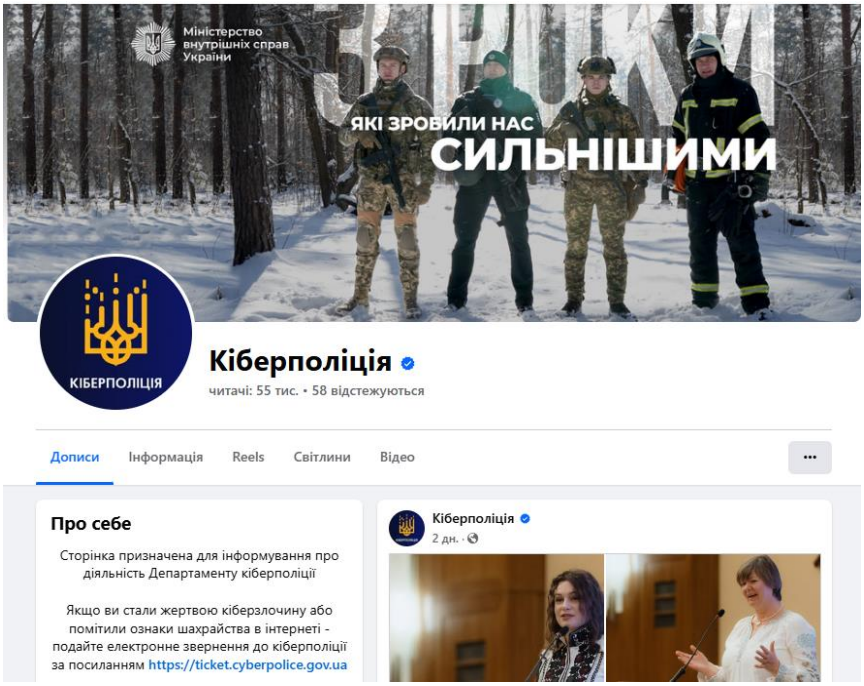


Рис. 3. Офіційна сторінка кіберполіції у Facebook

Рис. 4. Офіційна сторінка кіберполіції Cyberpolice Ukraine в X

Постановка проблеми. З початком повномасштабної війни в Україні різко зросла кількість кібератак на державні установи, критичну інфраструктуру та громадян. Оперативне інформування про такі кіберінциденти стало надзвичайно важливим, адже кібератаки можуть бути частиною гібридної війни та спричиняти дезорганізацію і паніку. ЗМІ, які ще до початку цього десятиліття вважалися головним джерелом оповіщення громадян державою, стали менш ефективними і популярними серед молоді та часто не встигають вчасно повідомляти про нові загрози або не мають достатньої технічної експертизи для детального висвітлення. Натомість соціальні мережі та месенджери стали альтернативним каналом для швидкого донесення інформації про кіберзагрози до різних аудиторій, проте цей напрямок все ще потребує розвитку.

Проблема полягає в тому, щоб проаналізувати, як саме джерела подають інформацію про кіберінциденти (швидкість, структура, наявність порад, орієнтація на аудиторію) та яким чином можна покращити ефективність роботи оповіщення населення про кіберінциденти через соціальні мережі і месенджери.

Результати та обговорення. Проте, існуючі методи інформування населення про кіберінциденти через соціальні мережі та месенджери не завжди повною мірою відповідають зростаючим вимогам користувачів. Для забезпечення своєчасного та всеосяжного інформування пропонується використання покращеної моделі оповіщення, яка має на меті інтегрувати сильні сторони існуючих практик та додати нові елементи для максимального охоплення, своєчасності та дієвості.

Пріоритезація оповіщень. Насамперед рекомендуємо запровадити інтуїтивно зрозумілу для населення класифікацію загроз, які виникають внаслідок кіберінцидентів. Вона полягатиме у поділі за пріоритетністю на червоний, помаранчевий та жовтий рівні загрози.

Таблиця 1.

Система оповіщень із пріоритезацією

Рівень Загрози	**Опис**	**Механізми Сповіщення**
Червоний	Для інцидентів із негайним, широкомасштабним впливом (наприклад, атаки на критичну банківську інфраструктуру, енергетику, державні сервіси першочергової важливості).	1. Миттєві push-сповіщення через мобільні мережі (аналогічно до сигналів повітряної тривоги, якщо технічно можливо та інфраструктурно підтримується). 2. Обов'язкові сповіщення через ключові державні та приватні мобільні додатки (наприклад, "Дія", банківські додатки, додатки комунальних служб). 3. Оповіщення через офіційні канали та чат-бот у Telegram.
Помаранчевий	Для значних загроз, що потенційно можуть мати широкий вплив або спрямовані на конкретні великі групи населення/сектори.	1. Пріоритетне розміщення на офіційних каналах у Telegram. 2. Оперативне інформування всіх ЗМІ.
Жовтий	Загальні попередження про нові схеми шахрайства, поради з кібергігієни, менш критичні технічні вразливості.	1. Публікації на офіційних сторінках у соцмережах. 2. Співпраця зі ЗМІ для регулярних рубрик.

Відповідно до поданої інформації про рівень загрози користувачі зможуть швидко оцінити важливість повідомлення, що надійшло.

Покращена структура повідомлення. Для того, щоб оперативно та результативно інформувати населення про кіберінциденти в соціальних мережах рекомендуємо

дотримуватися чіткої структури повідомлення. Таке повідомлення повинне включати наступні обов'язкові елементи (рис.5).

Рис. 5. Покращена структура повідомлення

Дотримання наведеної структури забезпечує ефективне інформування про кіберінциденти, що зменшує рівень паніки серед населення і підвищує загальний рівень обізнаності щодо поточних кіберризиків.

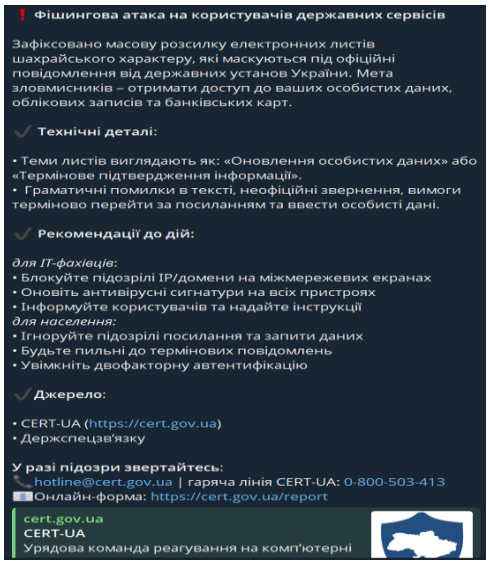


Рис. 6. Приклад повідомлення відповідно до запропонованої покращеної структури

Чат-бот в Telegram для персоналізації. Щоб забезпечити комфортні умови користувачам і дати їм можливість отримувати лише актуальні особисто для них загрози, можна створити окремий чат-бот у Telegram (як ключовому каналі для миттєвого поширення інформації). Він забезпечить механізм вказування особою її сфер інтересів, до прикладу, "IT-фахівець", "батько/мати", "власник бізнесу", "частий відвідувач пошти", "частий пасажир укрзалізниці", "клієнт певного банку" чи "користувач певного мобільно оператора". Завдяки цьому користувачі отримуватимуть попередження, що

враховують специфіку їхньої діяльності та вразливості, з якими вони найімовірніше можуть зіткнутися.

Для швидкого аналізу нових загроз, ідентифікації потенційно вразливих груп населення та автоматизованої адаптації рекомендацій для різних категорій користувачів можна застосувати штучний інтелект. Окрім того, ШІ можна використати для формулювання чітких, зрозумілих інструкцій різними мовами, що суттєво підвищить ефективність комунікації та рівень довіри серед усіх верств населення.

Додатковий спрощений механізм зворотного зв'язку. Ми пропонуємо інтегрувати кнопку "Повідомити про кіберінцидент" у “Дію”, яка є основним мобільним додатком України. Це дозволить громадянам швидко та легко звертатися до Кіберполіції, Держспецзв’язку або CERT-UA, що не лише пришвидшить реагування на інциденти, але й створить у громадян відчуття залученості та партнерства у забезпеченні національної кібербезпеки.

Інтерактивне навчання. В умовах стрімкого зростання кількості та складності кіберзагроз ключовим елементом модернізованої стратегії оповіщення про кіберінциденти стає впровадження інтерактивного навчання з питань кібербезпеки.

Пріоритетним напрямком діяльності стане впровадження інтерактивних навчальних кампанії через доступні цифрові канали. Громадян можна залучити до квізів, симуляцій фішингових атак та навчальних ігор через мобільні додатки та освітні платформи. Систематичне проведення вікторин і тестів у соцмережах та через чат-боти з елементами гейміфікації (рейтинги, досягнення, винагороди) стимулюватиме користувачів регулярно вдосконалювати знання з кібергігієни у захоплюючій формі.

Прямі ефіри з фахівцями Держспецзв'язку та Кіберполіції дозволять громадянам отримувати кваліфіковані консультації та підвищують довіру до державних інституцій. Запроваджена система анонімного збору інформації про підозрілі кіберінциденти від користувачів з подальшим експертним аналізом та публікацією рекомендацій забезпечить оперативне реагування на нові вектори атак.

Візуальний контент. Більш зрозумілим для звичайних користувачів способом донесення інформації є візуальний контент, і хоча його важче створити ніж звичайне текстове повідомлення, та він може значно підвищити ефективність оповіщення. До інструментів візуалізації відносять наприклад: інфографіку та короткі анімовані відеоролики. За рахунок більшої привабливості та пояснення на прикладах можна досягти більших охоплень і зацікавлення аудиторії в захисті від кіберзагроз.

Рис. 7. Постер у форматі інфографіки, для Telegram-каналу кіберполіції

Залучення інфлюенсерів та лідерів думок. Ефективність оповіщення про кіберінциденти значною мірою залежить від охоплення аудиторії та рівня довіри до поширюваної інформації. З метою розширення аудиторії рекомендуємо співпрацювати з громадськими діячами, популярними блогерами, а також місцевими лідерами. Це дозволить підвищити довіру до офіційних повідомлень і рекомендацій.

Також доцільним є розповсюдження інформації через великі регіональні та тематичні групи у Facebook, Telegram-канали ОСББ, батьківські чати та інші подібні об'єднання, оскільки це дозволить донести релевантну інформацію до цільових груп користувачів, враховуючи їхні специфічні потреби та контекст.

Висновки. Представлена вдосконалена модель інформування суспільства про кіберінциденти дозволить забезпечити вищу оперативність розповсюдження повідомлень на більш широку аудиторію. Вона забезпечить прямі сповіщення через мережу та інтеграцію з інформаційними системами ЗМІ, що дозволить досягнути швидшого донесення даних порівняно з традиційними способами інформування.

Пріоритезація та персоналізація оповіщень з цільовими рекомендаціями зробить повідомлення більш релевантними і ефективніше спонукатиме користувачів до конкретних дій. Водночас диверсифікація каналів комунікації знизить залежність від одного джерела, що підвищить загальну стійкість системи. Активне залучення користувачів через інтерактивні формати сприятиме кращому засвоєнню інформації і формуванню культури кібербезпеки. Залучення інфлюенсерів та спільнот допоможе значно розширити охоплення аудиторії, а чіткі інструкції і спрощені механізми повідомлення про інциденти дадуть змогу громадянам швидше реагувати на кіберзагрози.

Отже, комплексний підхід до інформування через соціальні мережі і месенджери дозволить створити більш оперативну, надійну і довірену систему комунікації, яка є важливою складовою зміцнення кібербезпеки України в сучасних умовах.

Список літератури

1. Гайдук О. В., Звєрєв В. П. Аналіз кіберзагроз в умовах стрімкого розвитку інформаційних технологій. *Кібербезпека: освіта, наука, техніка.* 2024. Т.3, №23. С. 225-236.
2. Ciekanowski Z, Żurawski S. Cyber Threat Analysis (CTA) in Current Conflicts. *IgMin Research*. 2024. Т.2 №4. С. 224-227.
3. Никончук Н.С., Маслова О.О. Кіберзлочинність в Україні: виклики сучасності. *Юридичний науковий електронний журнал*. 2021. №9. 202-205.
4. Худолій А. О. Кібербезпека: сучасні виклики перед Україною. *Acta De Historia & Politica: Saeculum XXI*. 2020. С. 138–146.
5. Держспецзв'язку. Атака на «Укрзалізницю» була актом кібертероризму. URL: https://glavcom.ua/country/incidents/ataka-na-ukrzaliznitsju-bula-aktom-kiberterorizmu-derzhspetszvjazku--1052297.html
6. Sud.ua. Онлайн-шахрайство під час війни: як захистити себе від фінансових втрат. URL: https://sud.ua/uk/news/ukraine/299885-onlayn-moshennichestvo-vo-vremya-voyny-kak-zaschitit-sebya-ot-finansovykh-poter#google_vignette
7. Brody-Освіта. Кіберполіція попереджає про шахрайство. Будьте обережні. URL: https://brody-osvita.gov.ua/kiberpoliciya-poperedzhae-13-35-49-21-06-2023/#
8. Провсе. Telegram, YouTube, Facebook, TikTok, Viber, Instagram чи Twitter: звідки українці дізнаються новини? URL: https://provse.te.ua/2024/08/telegram-youtube-facebook-tiktok-viber-instagram-chy-twitter-zvidky-ukraintsi-diznaiutsia-novyny/
9. Опора: найпопулярнішими соцмережами в Україні є телеграм, ютуб та фейсбук. URL: https://ms.detector.media/sotsmerezhi/post/35557/2024-07-16-opora-naypopulyarnishymy-sotsmerezhamy-v-ukraini-ie-telegram-yutub-ta-feysbuk/

10. Данько-Сліпцова А. А. Особливості Telegram як інтернет-платформи поширення соціально значущої інформації. *Український інформаційний простір*. 2023. Т.2 №12. С. 275-286 URL: http://ukrinfospace.knukim.edu.ua/article/view/291193

DEVELOPMENT OF AN IMPROVED METHODOLOGY FOR NOTIFYING THE PUBLIC ABOUT CYBER INCIDENTS

O.I. Harasymchuk, T.A. Kostyshyn, O.O. Liubchyk, M.E. Shved

Lviv Polytechnic National University
12, Stepan Bandera St., Lviv, 79013, Ukraine
Emails: oleh.i.harasymchuk@lpnu.ua, tetiana.kostyshyn.kb.2021@lpnu.ua,
olha.liubchyk.kb.2021@lpnu.ua, mariia.y.shved@lpnu.ua

In the context of the active implementation of digital technologies into public life and the consequent rise in the number of cyber threats, especially amidst the full-scale war in Ukraine, the issue of effectively informing the public about cyber incidents acquires critical importance. Traditional mass media are not always able to provide the necessary speed and reach for timely notification, making social networks and messengers key communication channels under current conditions. This article examines existing public notification systems for cyber incidents and, based on an analysis, proposes an improved methodology for disseminating information to the public. The main goal of this research is to develop an improved system for notifying the public about cyber incidents using social networks and messengers, which are currently the most effective platforms for rapid information dissemination. The research complements existing communication models in the context of cyber threats with new elements of digital interaction and increased user participation. The proposed recommendations can be applied by state structures to improve notification mechanisms through the use of an improved message structure, Telegram chatbots, visualization, interactive learning, additional feedback channels, and the involvement of influencers. The research presented in the article is qualitative in nature and is based on a comparative analysis of informational messages on social networks from state institutions. As a result of the research work, an improved model for informing about cyber incidents has been proposed, using a comprehensive, innovative approach adapted to Ukrainian conditions. In the future, it is advisable to evaluate the effectiveness of the new model after its implementation based on a comparison of the achieved results with those currently available.
Keywords: cyber incident, social networks, notification, cybersecurity, communication.

DOI 10.15276/imms.v15.no2.196
УДК 004.056.55

Informatics and Mathematical Methods in Simulation
Vol.15 (2025), No. 2, pp. 196-204

МЕТОД АФІННОГО ШИФРУВАННЯ НА ОСНОВІ КВАДРАТИЧНОЇ ФУНКЦІЇ

М.М. Касянчук[1], М.П. Голембйовський[2]

Західноукраїнський національний університет
11, Львівська вул., м. Тернопіль, 46009, Україна
Emails: kasyanchuk@ukr.net[1], mykhailo.2097@gmail.com[2]

Афінні шифри виникли ще в античні часи. Однак незважаючи на затвердження сучасних криптографічних стандартів, вони і на даний час зберігають певну актуальність у криптографії. Ці алгоритми можуть бути корисні, наприклад, при вивченні основ захисту інформації, для демонстрації принципів роботи шифрів заміни або при вивченні історії криптографії. Найбільш ефективними такі перетворення є у випадках, коли критичним параметром виступає час шифрування та передачі інформації. Афінні шифри ґрунтуються на лінійних математичних перетвореннях, які є вразливими до теперішніх криптоаналітичних атак. Тому метою даної роботи є розробка афінного шифру, в основі якого лежать квадратичні функції, усунення неоднозначності, яка виникає при розшифровуванні, та підвищення стійкості розробленого шифру до частотного аналізу за рахунок використання біграм. У роботі показано, що використання не лінійної, а квадратичної функції криптографічного перетворення дозволяє збільшити кількість варіантів ключа і, відповідно, криптографічну стійкість запропонованого алгоритму. Визначено умови, яким повинні відповідати коефіцієнти квадратного рівняння та значення модуля. Наведено покрокові шифруючі перетворення усіх символів розширеного українського алфавіту, формулу для розшифрування, а також відповідний приклад. Розроблено метод однозначного розшифровування квадратичних афінних шифрів за рахунок подвійної нумерації символів алфавіту. Це вдвічі збільшує значення модуля, однак дозволяє вибрати діапазон числових кодів, на яких квадратичні лишки будуть визначатися однозначно. Розроблено біграмний афінний шифр, який не піддається частотному аналізу одного символу алфавіту. Це підвищує криптостійкість методу, хоча збільшуються операнди, над якими виконуються арифметичні операції. Наведено приклад біграмного лінійного та квадратичного афінних шифрів.

Ключові слова: афінний шифр, лінійні перетворення, квадратична функція, криптографічні ключі, числові коди букв, біграмний шифр.

Вступ. Афінні шифри є одними з класичних методів симетричного шифрування [1, 2], які виникли ще в античні часи і використовувалися для захисту інформації [3]. Вони базуються на лінійних математичних перетвореннях над символами алфавіту [4, 5]. Їх використання навіть в епоху сучасних криптографічних стандартів має певну актуальність у криптографії [6, 7]. Афінні шифри можуть бути корисними в певних контекстах, попри свою слабкість порівняно з сучасними криптографічними методами [8, 9]. Зокрема, їх доцільно використовувати в навчальних курсах з криптографії для пояснення принципів роботи шифрів заміни і лінійних перетворень або при вивченні історичних аспектів криптографії.

До переваг афінних шифрів можна віднести простоту реалізації, швидкодію та лінійність математичних перетворень. Однак якщо зловмисник має достатньо зашифрованих повідомлень або хоча б одне зашифроване повідомлення і його відкритий текст, то цього може бути достатньо для розкриття ключа. Через лінійність шифру навіть частина зашифрованого тексту дозволяє зловмиснику встановити математичні зв’язки та відновити оригінальний текст. Оскільки афінний шифр є детермінованим і лінійним, то він не приховує статистичних властивостей тексту. Тому зловмисники можуть використати частотний аналіз для виявлення закономірностей та знаходження відповідностей у зашифрованому та відкритому текстах. Крім того, вимога взаємної

простоти елементів ключа та кількості букв в алфавіті значно обмежує кількість можливих варіантів ключів. Наприклад, для англійського алфавіту з 26-ти символів кількість допустимих взаємно простих чисел становить всього 12. Для зловмисника це робить досить реальною можливість перебору ключів навіть при невеликих за обсягом зашифрованих повідомленнях.

Отже, афінний шифр є вразливим до сучасних криптографічних атак, таких як атака методом грубої сили, атака за допомогою відомого відкритого тексту, атаки на основі математичних властивостей тощо. Афінний шифр є малоефективним при великих обсягах даних, тому його можна вважати безпечним тільки для досить обмежених ситуацій, в яких критичним параметром є час шифрування та передачі інформації.

Огляд літератури. Незважаючи на вказані недоліки, певна модифікація афінних шифрів (приховування кількості букв в алфавіті), їх поєднання з іншими криптографічними алгоритмами або деякими математичними перетвореннями (зокрема, використання системи залишкових класів [10, 11]) дозволяє підвищити стійкість зашифрованого повідомлення до криптоаналізу [12, 13].

Наприклад, для покращення захищеності баз даних в [14] після шифру Цезаря було запропоновано використовувати афінний шифр. Зашифрований текст зберігався у розділеному на поля текстовому файлі, що збільшувало його рівень безпеки.

У [15] розглядається алгоритм шифрування зображень, що базується на афінному шифрі. Спочатку шифруються позиції пікселів, потім афінний шифр використовується для розсіювання та заплутування зашифрованих значень. Такий підхід забезпечує кращу безпеку шифрування, а також високий рівень дифузії та заплутаності.

У [16] застосовано концепцію теорії афінних груп для захисту цифрових зображень на основі алгоритму DES і вейвлет-перетворень з використанням операцій множення матриць та векторного додавання.

Дослідження [17] описує метод, де спочатку використовується афінний шифр, а потім - шифр транспозиції. Це робить криптоаналіз більш складним завдяки використанню різних типів ключів та різній кількості символів у відкритому та зашифрованому текстах.

У [18] пропонується гібридна схема, що поєднує асиметричний криптоалгоритм Рабіна та симетричний афінний шифр. Афінне перетворення використовується для шифрування повідомлень, а система Рабіна — для шифрування та розшифрування ключів.

У [19] представлено варіант афінного шифру з використанням асиметричних ключів, які формуються з прямокутних матриць. Це рішення покращує криптостійкість запропонованого алгоритму шифрування.

В [20] описано можливість поєднання афінного шифру з генератором псевдовипадкових чисел Blum Blum Shub. Це дозволяє генерувати випадкові потоки ключів, що підвищує непередбачуваність зашифрованого тексту та збільшує криптостійкість системи.

Мета роботи. Метою даної роботи є розробка квадратичного афінного шифру, усунення неоднозначності при розшифровуванні та підвищення стійкості розробленого шифру до частотного аналізу за рахунок використання біграм. Для досягнення поставленої мети потрібно вирішити такі завдання:

- аналіз класичного афінного шифру, виявлення його переваг та недоліків, визначення необхідних арифметичних операцій, встановлення правил числового кодування тексту;

- розробка квадратичного афінного шифру, встановлення вимог до ключів шифрування/розшифрування, побудова таблиці з квадратичними криптографічними перетвореннями для розширеного українського алфавіту;

- розробка методу однозначного розшифровування для квадратичних афінних шифрів;

- розробка методу біграмного лінійного та квадратичного афінного шифрування.

Класичний афінний шифр. Класичне афінне шифрування полягає у лінійному перетворенні числового коду кожної букви у тексті. Це є частковий випадок для більш загального моноалфавітного шифру заміни і тому афінному шифру властиві усі вразливості даного класу шифрів. Зокрема, він може легко піддаватися частотному криптоаналізу, тобто володіє порівняно невеликою криптографічною стійкістю.

Найпростішим способом числового кодування тексту є заміна кожної букви її номером в алфавіті. В результаті отримається ряд 0, 1, 2, …, n-1, де n – кількість букв в алфавіті. Не враховуючи великих та малих символів та пробілів, відповідність між буквами українського алфавіту та їх числовими значеннями можна представити таблицею 1.

Таблиця 1.

Відповідність між буквами українського алфавіту та числами

Буква	а	б	в	г	ґ	д	е	є	ж	з	и
Номер	00	01	02	03	04	05	06	07	08	09	10
Буква	і	ї	й	к	л	м	н	о	п	р	с
Номер	11	12	13	14	15	16	17	18	19	20	21
Буква	т	у	ф	х	ц	ч	ш	щ	ь	ю	я
Номер	22	23	24	25	26	27	28	29	30	31	32

Далі, згідно властивостей модульної арифметики, відбувається шифруюче лінійне перетворення кожного числа, яке відповідає символу відкритого тексту, за такою формулою:

$$X=(a\cdot x+s) \bmod n, \quad (1)$$

де x і X – номери букв відкритого та зашифрованого тексту відповідно; пара a ключі шифру, для яких повинні виконуватися умови: $1\le a\le n$-1, НСД (a, n)=1, $0\le s\le n$-1.

Для розшифровування використовується аналогічне перетворення:

$$x=(A\cdot X+S) \bmod n, \quad (2)$$

де $A=a^{-1} \bmod n$ – обернений елемент до числа a за взаємнопростим модулем n; $S=(-As) \bmod n$.

В таблиці 2, згідно виразів (1)-(2), наведено приклад криптографічних перетворень повідомлення «бот» (01 18 22 згідно таблиці 1) з ключем шифрування a=13, s=9. Відповідно, ключ для розшифровування $A=13^{-1} \bmod 33=28$, $S=(-28\cdot 9) \bmod 33=(5\cdot 9) \bmod 33=12$.

Таблиця 2.

Приклад криптографічних перетворень за допомогою афінних шифрів

Зашифрування				Розшифрування			
Повідомлення	б	о	т	Шифртекст	т	ї	ю
x	01	18	22	X	22	12	31
$13\cdot x+9$	22	243	295	$28\cdot x+12$	628	348	880
$X=(13\cdot x+9) \bmod 33$	22	12	31	$x=(28\cdot x+12) \bmod 33$	01	18	22
Шифртекст	т	ї	ю	Повідомлення	б	о	т

Якщо a=1, то афінний шифр трансформується у шифр зсуву (або шифр Цезаря) і процеси шифрування та розшифровування зводяться до простого лінійного зсуву:

$$X=(x+s) \bmod n,\ x=(X+S) \bmod n, \quad (3)$$

де $S=(-s) \bmod n=n-s$.

Якщо $a\neq 1$ і s=0, то шифрування та розшифровування здійснюються тільки за допомогою операції множення:

$$X=(ax) \bmod n,\ x=(AX) \bmod n, \tag{4}$$

Якщо зловмиснику відома потужність алфавіту n, то швидкий криптоаналіз можливий навіть прямим перебором всіх можливих варіантів. Наприклад, для українського алфавіту n=33, параметр a можна вибрати 20-ма способами (функція Ейлера $\phi(33)=20$), параметр – 33-ма способами. Тобто для успішного криптоаналізу лінійного афінного шифру потрібно розглянути всього 660 варіантів ключа, причому обчислення будуть виконуватися над порівняно невеликими операндами. Крім того, оскільки однаковим буквам відкритого повідомлення відповідають однакові букви шифртексту, то афінні шифри вразливі і до частотного аналізу.

Квадратичні криптографічні перетворення. Збільшити кількість варіантів ключа дозволяє використання не лінійної, а квадратичної функції криптографічного перетворення:

$$X=\left(b\cdot x^2+c\cdot x+d\right)\bmod n=\left(b\cdot\left(\left(x+\frac{c}{2\cdot b}\right)^2+\frac{d}{b}-\frac{c^2}{4\cdot b^2}\right)\right)\bmod n, \tag{5}$$

де b, c, d – коефіцієнти квадратичної функції, які повинні бути менші за n.

Після перетворення $X=\left(b\cdot\left(\left(x+\left(c\cdot\left((2\cdot b)^{-1}\bmod n\right)\right)\bmod n\right)^2+\right.\right.$ $\left.\left.+\left(d\cdot\left(b^{-1}\right)\bmod n-c^2\cdot\left((2\cdot b)^{-2}\right)\bmod n\right)\bmod n\right)\right)\bmod n$ вираз (5) доцільно переписати у такому вигляді:

$$X=\left(b\cdot\left((x+f)^2+g\right)\right)\bmod n, \tag{6}$$

де $f=\left(c\cdot\left((2\cdot b)^{-1}\bmod n\right)\right)\bmod n;\ g=\left(d\cdot\left(b^{-1}\right)\bmod n-c^2\cdot\left((2\cdot b)^{-2}\right)\bmod n\right)\bmod n$.

Звідси випливає, що параметри b і $2b$, для яких потрібно шукати обернені елементи за модулем n, повинні бути взаємно прості з кількістю букв в алфавіті. Для автоматичного виконання таких умов доцільно потужність алфавіту розширити до деякого простого числа. В цьому випадку половина елементів зведеної системи лишків будуть квадратичними лишками.

Розшифрування повинно відбуватися в зворотному порядку і передбачає обчислення кореня квадратного за модулем, методи пошуку якого описані, наприклад, в [21, 22]. Отже, з формули (6) можна отримати:

$$x=\left(\sqrt{b^{-1}X-g}-f\right)\bmod n, \tag{7}$$

Запропоноване квадратичне перетворення ускладнює криптоаналіз в порівнянні з лінійним афінним шифром, особливо, якщо зловмиснику невідома потужність алфавіту, однак не усуває вразливості до частотного аналізу. Крім того, розв'язок квадратного рівняння або пошук квадратного кореня приводить до неоднозначності при розшифруванні

Для прикладу розглянемо функцію шифрування:

$$X=(3x^2+7x+11) \bmod 37. \tag{8}$$

Тут український алфавіт розширений з 33 до 37 символів додаванням, наприклад, крапки (.) - номер 33, коми (,) – номер 34, пробілу ( ) – номер 35 та знаку оклику (!) – номер 36.

Згідно формули (6) b=3, інші параметри шукаються таким чином:

$$f=\left(\frac{c}{2b}\right)\bmod 37=\frac{7}{6}\bmod 37=(7\cdot 31)\bmod 37=32\bmod 37=-5\bmod 37;$$

$$g=\left(\frac{d}{b}-\left(\frac{c}{2b}\right)^2\right)\bmod 37=\left(\frac{11}{3}-\left(\frac{7}{6}\right)^2\right)\bmod 37=\left(11\cdot 25-(7\cdot 31)^2\right)\bmod 37=$$

$$=28\bmod 37=-9\bmod 37.$$

Тоді з виразу (8) отримується:

$$X=\left(3\cdot\left((x-5)^2-9\right)\right)\bmod 37. \tag{9}$$

В таблиці 3 наведено приклади покрокових шифруючих перетворень усіх символів алфавіту потужністю 37.

Таблиця 3

Квадратичні афінні перетворення над символами розширеного українського алфавіту

x	00	01	02	03	04	05	06	07	08	09	10	11	12
$x^2 \bmod 37$	0	1	4	9	16	25	36	12	27	7	26	10	33
$(x\text{-}5)^2 \bmod 37$	25	16	9	4	1	0	1	4	9	16	25	36	12
$((x\text{-}5)^2\text{-}9) \bmod 37$	16	7	0	32	29	28	29	32	0	7	16	27	3
$3\cdot((x\text{-}5)^2\text{-}9) \bmod 37$	11	21	0	22	13	10	13	22	0	21	11	7	9
x	13	14	15	16	17	18	19	20	21	22	23	24	25
$x^2 \bmod 37$	21	11	3	34	30	28	28	30	34	3	11	21	33
$(x\text{-}5)^2 \bmod 37$	27	7	26	10	33	21	11	3	34	30	28	28	30
$((x\text{-}5)^2\text{-}9) \bmod 37$	18	35	17	1	24	12	2	31	25	21	19	19	21
$3\cdot((x\text{-}5)^2\text{-}9) \bmod 37$	17	31	14	3	35	36	6	19	1	26	20	20	26
x	26	27	28	29	30	31	32	33	34	35	36		
$x^2 \bmod 37$	10	26	7	27	12	36	25	16	9	4	1		
$(x\text{-}5)^2 \bmod 37$	34	3	11	21	33	10	26	7	27	12	36		
$((x\text{-}5)^2\text{-}9) \bmod 37$	25	31	2	12	24	1	17	35	18	3	27		
$3\cdot((x\text{-}5)^2\text{-}9) \bmod 37$	1	19	6	36	35	3	14	31	17	9	7		

Для прикладу, слово «вежа», яке має числовий код (02 06 08 00), перетвориться у шифртекст (00 13 00 11) або «айаі» згідно таблиці 1. Букви «в» і «ж» шифруються однаковим символом «а». Це якраз і пов'язується з тим, що квадратне рівняння має два розв'язки і результат розшифрування згідно формули (7) буде неоднозначний. Зокрема, враховуючи, що $3^{-1} \bmod 37=25$, хід розшифрування буде такий:

1) $\left(\sqrt{3^{-1}\cdot 0+9}+5\right)\bmod 37=(\pm 3+5)\bmod 37=2$ і 8, що відповідає буквам «в» і «ж»;

2) $\left(\sqrt{3^{-1}\cdot 13+9}+5\right)\bmod 37=\left(\sqrt{25\cdot 13+9}+5\right)\bmod 37=(\pm 1+5)\bmod 37=4$ і 6, що відповідає буквам «ґ» і «е»;

3) $\left(\sqrt{3^{-1}\cdot 11+9}+5\right)\bmod 37=\left(\sqrt{25\cdot 11+9}+5\right)\bmod 37=(\pm 5+5)\bmod 37=0$ і 10, що відповідає буквам «а» і «и».

Метод однозначного розшифровування квадратичних афінних шифрів. Зрозуміло, що неоднозначність при розшифровуванні є суттєвим недоліком квадратичних афінних шифрів, оскільки потрібно розглядати 2^k варіантів, де k – кількість букв у повідомленні. Усунути його дозволяють властивості таблиці 3. Видно, що при використанні функції $X=x^2 \bmod n$ результати шифрування симетричні відносно параметра $\frac{n}{2}$. Для функції $X=(x+f)^2 \bmod n$ усі значення X циклічно зсуваються на величину f. Зсув відбувається ліворуч при $f>0$ і праворуч при $f<0$. При використанні двох інших функцій відбувається зміна результату шифрування в залежності від параметрів g і b. Це означає, що при $f<0$ на ділянці від $x=f$ до $x=\frac{n-1}{2}+f$, а при $f>0$ – від $x=\frac{n-1}{2}-f$ до $x=n-f$ квадратичні лишки повторюватися не будуть. Тому якщо букви відкритого тексту вибирати на цій ділянці, то результат розшифрування буде однозначний. Щоб охопити весь алфавіт його символи потрібно пронумерувати двічі, кількість символів доцільно вибрати так, щоб число $2n$-1, яке визначає потужність алфавіту, було простим.

В таблиці 4 наведено фрагмент алфавіту потужністю 19 з перших 10-ти букв з функцією шифрування $X=(x^2+6x+7) \bmod 19=((x+3)^2\text{-}2) \bmod 19$. Для шифрування і

розшифровування вибирається діапазон від x=07 (буква «є») до x=16 (буква «е»). Шифртекст слова «вежа» - (12 16 08 10) буде «ґєдд» - (14 17 05 15).

Враховуючи, що b=1, процедура розшифрування матиме такий вигляд:

1) $\left(\sqrt{14+2}-3\right)\text{mod}19=(\pm4-3)\,\text{mod}19=1$ і 12;

2) $\left(\sqrt{17+2}-3\right)\text{mod}19=(0-3)\,\text{mod}19=16$;

3) $\left(\sqrt{5+2}-3\right)\text{mod}19=(\pm8-3)\,\text{mod}19=5$ і 8;

4) $\left(\sqrt{15+2}-3\right)\text{mod}19=(\pm6-3)\,\text{mod}19=3$ і 10.

Вибравши числа з діапазону від 7 до 16, можна однозначно отримати розшифрований числовий код (12 16 08 10), що відповідає слову «вежа».

Таблиця 4

Однозначне розшифрування для фрагменту алфавіту

Буква	а	б	в	г	ґ	д	е	є	ж	з
x	00	01	02	03	04	05	06	07	08	09
x^2mod 19	0	1	4	9	16	6	17	11	7	5
$((x+3)^2-2)$mod 19	7	14	4	15	9	5	3	3	5	9
Буква	а	б	в	г	ґ	д	е	є	ж	з
x	10	11	12	13	14	15	16	17	18	19
x^2mod 19	5	7	11	17	6	16	9	4	1	0
$((x+3)^2-2)$mod 19	15	4	14	7	2	18	17	18	2	7

Біграмний афінний шифр. Усі розглянуті вище афінні шифри вразливі до частотного аналізу, оскільки тим самим буквам відкритого тексту відповідають однакові букви зашифрованого. Усунути цей недолік дозволяє використання шифрів складної заміни, зокрема біграмних, в яких блоком шифрування виступають пари букв – біграми. У відкритому повідомленні кількість символів має бути парною (непарну кількість можна доповнити крапкою). Модуль криптографічного перетворення, згідно розширеної таблиці 1, повинен перевищувати 3236 (повідомлення «я!») і бажано, щоб був простим числом. Передача зашифрованого повідомлення буде здійснюватися в числовому коді.

Нехай потрібно зашифрувати розбите на біграми повідомлення «шифри.» з числовим кодом (2810 2420 1033). Для прикладу у лінійному випадку вибираємо такі ключі шифрування: a=31, s=1590, n=3701. Відповідно формула шифрування: $X=(31\cdot x+1590)$ mod 3701. Обчисливши ключі для розшифрування $A=31^{-1}$ mod 3701=2149, $S=(-2149\cdot1590)$ mod 3701=2814, можна отримати формулу для розшифрування: $x=(2149\cdot X+2814)$ mod 3701. Для квадратичного шифрування можна вибрати розглянуте вище рівняння з модулем n=3701: $X=(x^2+6x+7)\text{mod }3701=((x+3)^2-2)\text{mod }3701$. Розшифрування відбувається за допомогою формули $x_{1,2}=\left(\pm\sqrt{X+2}-3\right)\text{mod }3701$.

У таблиці 5 наведено результати шифрування і розшифровування біграмним лінійним та квадратичним афінними шифрами.

Таблиця 5.

Результати шифрування і розшифровування біграмним лінійним та квадратичним афінними шифрами

Зашифрування				Розшифрування			
Повідомлення	ши	фр	и.	Лінійний афінний шифр			
x	2810	2420	1033	X	3577	2590	304
Лінійний афінний шифр				$2149\cdot X+2814$	7689787	5568724	656110
$31\cdot x+1590$	88700	76610	33613	x	2810	2420	1033
X	3577	2590	304	Повідомлення	ши	фр	и.
Квадратичний афінний шифр				Квадратичний афінний шифр			

Зашифрування				Розшифрування			
x^2+6x+7	7912967	5870927	1073294	X	229	1141	4
X	229	1141	4	$x_{1,2}$	885, 2810	1275, 2420	1033, 2662
				Повідомлення	2810, ши	2420, фр	1033, и.

З таблиці 5 видно, що букві «и», яка два рази міститься у відкритому повідомленні, у шифртексті відповідають зовсім різні числові коди. Тому частотний аналіз по одній букві в даному випадку непридатний, що підвищує криптостійкість методу. Однак збільшуються операнди, над якими виконуються арифметичні операції.

Крім того, для квадратичного шифру знову виникає проблема неоднозначності. Вибір букв у певному діапазоні, як було показано на прикладі таблиці 4, в цьому випадку непридатний. Однак, згідно таблиці 1, для розв'язків (8 85), (12 75), (26 62) не існує відповідної букви, що значно спрощує пошук правильного розв'язку.

Висновки. Дана робота присвячена розробці афінного шифру, в основі якого лежить використання не лінійної, а квадратичної функції криптографічного перетворення. Це дозволяє збільшити кількість варіантів ключа і, відповідно, криптографічну стійкість запропонованого методу. Визначено умови, яким повинні відповідати коефіцієнти квадратного рівняння та значення модуля. Наведено покрокові шифруючі перетворення усіх символів розширеного українського алфавіту, формулу для розшифрування, а також відповідний приклад. Розроблено метод однозначного розшифровування квадратичних афінних шифрів за рахунок подвійної нумерації символів алфавіту. Це вдвічі збільшує значення модуля, однак дозволяє вибрати діапазон числових кодів, на яких квадратичні лишки будуть визначатися однозначно. Розроблено біграмний афінний шифр, який не піддається частотному аналізу одного символу алфавіту. Це підвищує криптостійкість методу, хоча збільшуються операнди, над якими виконуються арифметичні операції. Наведено приклад біграмного лінійного та квадратичного афінних шифрів.

Список літератури

1. Корченко О.Г., Сіденко В.П., Дрейс Ю.О. Прикладна криптологія: системи шифрування: підручник. К.: ДУТ, 2014. 448 с.
2. Глинчук Л.Я. Криптологія: навч.-метод. посіб. Луцьк: Вежа-Друк, 2014. 164 с.
3. Гребенніков В.В. Історія криптології та секретного зв'язку. К.: КНТ, 2023. 800 с.
4. Кожухівський А.Д., Горбенко І.Д., Гайдур Г.І., Кожухівська О.А., Марченко В.В. Математичні методи криптології. К.: ДУТ, 2021. 244 с.
5. Гапак О.М., Балога С.І. Захист інформації в комп'ютерних системах: підручник. Ужгород: УжНУ, 2021. 184 с.
6. Красиленко В.Г., Грабовляк С.К. Матричні афінні шифри для створення цифрових сліпих підписів на текстографічні документи. *Системи обробки інформації*. Х.: ХУПС, 2011. Вип. 7(97). С. 60 – 63.
7. Красиленко В.Г., Нікітович Д.В. Удосконалення та моделювання матричних афінних шифрів для криптографічних перетворень зображень. *Електроніка та інформаційні технології*: збірник наукових праць. Львів: Львівський національний університет імені Івана Франка, 2017. Вип. 7. С. 20-42.
8. Щур Н.О., Покотило О.А. Основи криптології: навч. посібник. Житомир: Державний університет «Житомирська політехніка», 2021. 120 с.
9. Гапак О.М. Криптоаналіз. Криптографічні протоколи: навчальний посібник. Ужгород: УжНУ, 2021. 93 с.
10. Касянчук М.М. Теорія та математичні закономірності досконалої форми системи залишкових класів. *Питання оптимізації обчислень* (*ПОО–XXXV*): Праці Міжнародного симпозіуму. Київ–Кацивелі. 2009. Т.1. С. 306–310.
11. Kasianchuk M. Conception of theoretical bases of the accomplished form of Krestenson's transformation and its practical application. *Advanced Computer Systems and Network: Design*

and Application (*ACSN–2009*): Proceedings of the 4–th International Conference. L'viv. 2009. P.299–301.
12. Kasianchuk M., Yakymenko I., Nykolaychuk Y.M. Symmetric cryptoalgorithms in the residue number system. *Cybernetics and Systems Analysis*. Vol.57(2), 2021. P.329–336.
13. Nykolaychuk Y., Yakymenko I., Vozna Y., Kasianchuk M. Residue number system asymmetric cryptoalgorithms. *Cybernetics and Systems Analysis*. Vol.58(4), 2022. P.611–618.
14. Hammood D.A., Maitham A. Implementation and Enhancement Affine Cipher of Database. *Journal of Engineering and Sustainable Development.* Vol.20, 2016. P.264–276.
15. Ke Q., Liao Q.-N., Li A.-Q., Gao R. Digital Image Encryption Algorithm Based on Affine Cipher. In *Advances in Intelligent Systems and Computing*; Springer International Publishing: Cham, 2019. P. 578–585.
16. Soekarta R., Sigit M. Implementation of Affine Group Algebra on Digital Image Security. *Mob. Forensics*. Vol.4, 2023. P. 137–146.
17. Alhassan M.J., Hassan A., Sani S., Alhassan Y. A Combine Technique of an Affine Cipher and Transposition Cipher. *Journal of Research in Applied Mathematics*, Vol. 7, 2021. P. 8–12.
18. Budiman M.A., Handrizal H., Azzahra S. An Implementation of Rabin-p Cryptosystem and Affine Cipher in a Hybrid Scheme to Secure Text. *J. Phys. Conf. Ser.* Vol. 1898, 2021. P. 012042.
19. Maxrizal M., Aniska Prayanti B.D. Application of Rectangular Matrices: Affine Cipher Using Asymmetric Keys. *CAUCHY – Jurnal Matematika Murni dan Aplikasi*. Vol. 5(4), 2019. P. 181–185.
20. Shoup V. A Computational Introduction to Number Theory and Algebra. Cambridge University Press: Cambridge, England. 2009. 578 p.
21. Yakymenko I., Kasianchuk M., Shylinska I., Shevchuk R., Yatskiv V., Karpinski M. Polynomial Rabin Cryptosystem Based on the Operation of Addition. Proceedings of the 12th International Conference "Advanced Computer Information Technology (ACIT 2022)" (Ruzomberok, Slovakia), 2022. P. 345–350.
22. Ivasiev S., Kasyanchuk M., Yakymenko I., Gomotiuk O., Shylinska I., Bilovus L. Algorithmic Support for Rabin Cryptosystem Implementation Based on Addition. Proceedings of the 10th International Conference "Advanced Computer Information Technology (ACIT 2020)" (Deggendorf, Germany). 2020. P. 779-782

AFFINE ENCRYPTION TECHNIQUE BASED ON QUADRATIC FUNCTION

M.M. Kasianchuk[1], M.P. Holembiovskyi[2]

West Ukrainian National University
11, Lvivska Str., Ternopil, 46009, Ukraine
Emails: kasyanchuk@ukr.net[1], mykhailo.2097@gmail.com[2]

Affine ciphers originated in ancient times. However, despite the advent of modern cryptographic standards, they are still relevant in cryptography. These algorithms can be useful, for example, when teaching the fundamentals of data protection, demonstrating the principles of substitution ciphers, or studying the history of cryptography. Such transformations are particularly effective in cases where encryption and data transmission speed is a critical parameter. Affine ciphers are based on linear mathematical transformations that are vulnerable to current cryptanalytic attacks. Therefore, the purpose of this work is to develop an affine cipher based on quadratic functions, eliminate the ambiguity that arises during decryption, and increase the resistance of the developed cipher to frequency analysis using bigrams. The study demonstrates that using a quadratic rather than a linear cryptographic transformation function allows increasing the number of possible keys, thereby improving the cryptographic strength of the proposed algorithm. Conditions are defined for the coefficients of the quadratic equation and the modulus values. Step-by-step encryption transformations for all characters of the extended Ukrainian alphabet are given as well as the decryption formula, and an illustrative example. A technique for unambiguous decryption of quadratic affine ciphers is developed based on double numbering of the alphabet symbols. This leads to doubling the modulus value, but allows you to select a range of number codes for which quadratic residues can be uniquely determined. A bigram affine cipher is developed, which is resistant to single-character frequency analysis. This increases the cryptographic strength of the technique, although the operands on which arithmetic operations are performed increase. Examples of linear and quadratic affine cipher bigram are given.

Keywords: affine cipher, linear transformations, quadratic function, cryptographic keys, letter number codes, bigram cipher.

DOI 10.15276/imms.v15.no2.205
УДК 004.42: 004.021

Informatics and Mathematical Methods in Simulation
Vol.15 (2025), No. 2, pp. 205-217

СТРУКТУРНЕ ПОДАННЯ UML-МЕТАМОДЕЛІ У ФОРМАТІ JSON

Н.О. Комлева, М.І. Нікітченко

Національний університет «Одеська політехніка»
1, Шевченка пр., м. Одеса, 65044, Україна
Emails: komleva@op.edu.ua, maksym.nikitchenko@gmail.com

В роботі розглянуто проблему формального представлення моделей UML (Unified Modeling Language) у відкритому форматі, придатному для перевірки, трансформації та інтеграції в інженерні процеси розробки програмного забезпечення. Як концептуальну основу дослідження використано UML-метамодель, що включає два взаємопов'язані подання: структурне (визначає об'єкти та зв'язки моделі) та поведінкове (описує активності, стани й послідовності). Така метамодель слугує базою для інкрементального контролю консистентності програмної архітектури. У даній роботі особливу увагу приділено структурному представленню метамоделі UML, який є базовим для відображення архітектури системи у вигляді взаємопов'язаних типів об'єктів. Для цього представлення реалізовано формалізоване представлення ключових метасутностей UML, таких як класи, атрибути, асоціації, пакети, компоненти та об'єкти, у форматі JSON (JavaScript Object Notation). Обране подання базується на застосуванні стандарту JSON Schema, що дозволяє чітко описувати допустимі структури, типи даних, іменовані властивості та правила валідації. У результаті побудовано повноцінну множину типів і відношень, які визначають дозволену конфігурацію UML-моделі як структурованого набору об'єктів із чітко визначеними зв'язками між ними. Для підтримки синтаксичної і семантичної цілісності цієї моделі було сформульовано систему формалізованих інваріантів, яка фіксує критичні обмеження, зокрема, вимоги до ідентифікаторів, коректного вкладення елементів, суворої типізації атрибутів, а також відсутності циклічних залежностей у композиційних і успадкованих структурах. Це закладає основу для уніфікованого представлення структурних моделей та подальшої автоматизованої перевірки їх коректності. Частина правил реалізована засобами самої JSON-схеми, а для складніших передбачено виконання через зовнішні механізми перевірки. Запропонований підхід дозволяє забезпечити чітку відповідність між абстрактними об'єктами UML та їх представленням у відкритому форматі, що, в перспективі, відкриває можливість уніфікованої інтеграції з іншими засобами моделювання, CI/CD та інструментами генерації коду. Дослідження зосереджено на формалізмі, прозорості та узгодженості з UML-специфікацією, які є необхідними для підтримки інкрементального контролю консистентності із забезпеченням сумісності між складовими метамоделі.

Ключові слова: моделювання, UML, метамодель, JSON Schema, валідація, інваріанти.

Вступ. UML (Unified Modeling Language) є загальновизнаним стандартом для візуального моделювання програмних систем. У типовому випадку UML-модель охоплює кілька взаємодоповнювальних подань, серед яких найбільш поширеними є структурні – діаграми класів, компонентів, об'єктів, пакетів, що відображають статичну архітектуру системи – та поведінкові, які охоплюють динамічні аспекти функціонування через діаграми активностей, станів, послідовностей, комунікацій тощо. Додатково до них можуть використовуватись фізичні (розгортання), вимог (використання) та інші подання, що формують повну модель програмної системи.

У даній роботі розглянуто проблему формального представлення моделей UML у відкритому форматі, придатному для перевірки, трансформації та інтеграції в інженерні процеси розробки програмного забезпечення. В основі дослідження – метамодель UML, що складається зі структурного подання для фіксації об'єктів і зв'язків та поведінкового

для опису активностей і послідовностей [1], яка забезпечує інкрементальний контроль консистентності моделей.

У межах даної статті зосереджено увагу на структурному представлені, який є критично важливим для подальшої верифікації та узгодженості моделі системи. Особливу увагу приділено реалізації цього подання у форматі JSON (JavaScript Object Notation), який розглядається як альтернатива традиційному XMI (XML Metadata Interchange). Незважаючи на те, що XMI є офіційним стандартом OMG (Object Management Group), він часто виявляється надмірно складним, громіздким і малопридатним до інкрементальної обробки в CI/CD-середовищах.

Натомість JSON є легковагим текстовим форматом, що набув широкого використання в сучасних інструментах моделювання (наприклад, StarUML використовує .mdj, Modelio – .uml.json). Проте, відсутність єдиного формалізованого стандарту для JSON-подання UML обмежує можливість гарантованої перевірки моделей, перешкоджає автоматизації трансформацій та ускладнює вбудовування у сучасні інженерні процеси [2]. Саме тому виникає необхідність у створенні структурного подання метамоделі, здатної формально описати допустимі об’єкти, зв’язки та обмеження UML-моделі у вигляді строго визначених типів та правил, виражених у термінах JSON Schema.

Огляд існуючих рішень. Значну увагу дослідники приділяють забезпеченню консистентності UML/OCL (Object Constraint Language) моделей та перевірці інваріантів при інкрементальних змінах. У роботі [3] запропоновано метод логічної абстракції для інкрементальної верифікації, що дозволяє ефективно перевіряти часткові оновлення без повної повторної валідації. Систематизований огляд технік обрізання моделей та методів надання зворотного зв’язку з метою підвищення ефективності перевірки діаграм класів UML/OCL наведено у [4].

Паралельно з розвитком UML-моделювання поширюється використання формату JSON, який забезпечує сумісність із сучасними веб-орієнтованими технологіями. Офіційні специфікації JSON Schema для опису структури й обмежень JSON-документів, придатних для валідації UML-моделей, наведено в [5; 6]. Відповідність UML-структур формату JSON та правила їх кодування згідно з рекомендаціями OGC представлені у [7]. Проблеми формалізації та складності валідації JSON-схем, а також методи спрощення таких перевірок, детально розглядаються у [8; 9].

Окрім традиційних підходів до перевірки UML-моделей з використанням OCL, актуальними є також онтологічні методи. У [10] досліджено використання reasoner-інструментів для перевірки узгодженості класів, об’єктів та діаграм станів UML, що відкриває перспективи застосування семантичних технологій у моделюванні. Узагальнений огляд формальних і практичних методів забезпечення консистентності UML-класів подано в [11]. На практичному рівні ведуться спроби трансформувати JSON-схеми у UML-структури для задач зворотної інженерії, прикладом чого є реалізований конвертер JSONSchema-to-UML [12]. Актуальність теми підтверджується також документацією StarUML [13] та Modelio [14], де розкрито структуру відповідних форматів (.mdj, .uml.json), які зберігають UML-моделі на основі JSON.

У напрямі онтологічної трансформації UML-класів варто відзначити роботу [15], в якій запропоновано метод формалізованого перетворення UML-моделей з використанням онтологічних обмежень для подальшої перевірки їхньої коректності. Перспективи колаборативного UML-моделювання у хмарних середовищах та вимоги до взаємодії інструментів обговорюються в [16], де підкреслюється важливість формальних підходів для підтримки інтеграції. Нарешті, дослідження [17] акцентує на формалізації чотиришарової архітектури метамоделювання, що є основою для подальшого розвитку інструментів валідації UML-моделей з урахуванням рівнів абстракції.

Таким чином, аналіз літератури підтверджує, що поєднання формального представлення UML-моделей у форматі JSON з механізмами автоматизованої валідації є

перспективним напрямом, що потребує подальшого розвитку як на методологічному, так і на інструментальному рівнях.

Мета дослідження. Метою даного дослідження є побудова формалізованого структурного подання UML- метамоделі у форматі JSON, яка забезпечує однозначне відображення ключових метасутностей на елементи JSON Schema. Таке подання поєднує формальну строгість опису UML-структури з гнучкістю JSON-середовища, що створює підґрунтя для розробки інструментів автоматизованої валідації, трансформації та генерації коду на основі UML-моделей у відкритому форматі.

Формалізоване структурне подання UML-метамоделі у форматі JSON. У межах цього дослідження запропоновано структурне подання UML-метамоделі у форматі JSON, що задається у вигляді пари:

$$M_S^{JSON} = < T, R >, \quad (1)$$

де T – множина допустимих типів структурних метасутностей, відображених у форматі JSON; R – множина структурних відношень між об'єктами, які також представлені як обмеження на JSON-структури.

У контексті підтримуваних діаграм UML множина типів T включає:

T = {UMLClass, UMLAttribute, UMLOperation, UMLAssociation, UMLGeneralization, UMLEnumeration, UMLPackage, UMLComponent, UMLObject}.

Множина відношень R задає ключові структурні залежності між об’єктами:

- hasAttribute ⊆ UMLClass × UMLAttribute;
- hasOperation ⊆ UMLClass × UMLOperation;
- typedBy : UMLAttribute → (UMLClass ∪ Primitive);
- generalizes ⊆ UMLClass × UMLClass;
- associates ⊆ UMLAssociation × UMLClass.

Таким чином, кожна діаграма UML (класи, компоненти, об’єкти, пакети) моделюється як часткове відображення цієї загальної структури M_S^{JSON}, з відповідними обмеженнями на типи, відношення та інваріанти, що накладаються через JSON Schema.

Кожен об'єкт o∈T у структурному поданні описується як структурований JSON-об’єкт, що має фіксований набір атрибутів, такі як ідентифікатор (поле id), ім’я (поле name), вкладені елементи або посилання (наприклад, attributes, associations, dependencies), типова специфікація (наприклад, type, superClassId, interface). Крім того, на множину елементів застосовуються валідаційні інваріанти $\Sigma=\{\sigma_1,\ldots,\sigma_n\}$, які формалізують правила структурної коректності моделі.

Вимоги до структурного подання UML у JSON. Для формальної обробки UML-моделей у JSON-форматі необхідно задати чіткі вимоги, які гарантують коректність, уніфікованість, інтероперабельність та розширюваність моделі.

Основні вимоги до структурного подання, що охоплює статичні аспекти (класи, атрибути, зв’язки, пакети):

- Метасутності: підтримка базових елементів UML, таких як Class, Attribute, Operation, Association, Package, Enumeration, Interface. Описуються у JSON через вкладені об'єкти;
- Ідентифікація об’єктів: кожен елемент повинен мати унікальний id, що дозволяє здійснювати посилання між елементами;
- Цілісність структури: заборонено цикли композиції та успадкування; граф структури має бути ациклічним. Визначається на етапі валідації графу моделі;
- Читабельність та логічність: імена полів, типів та вкладених елементів мають бути послідовними, стандартизованими, зрозумілими. Наприклад, name, visibility, type;
- Розширюваність (стереотипи): передбачена підтримка метаінформації: стереотипів, тегів, профільних атрибутів (якщо потрібно);

– Формат зв’язків: всі взаємозв’язки реалізуються через посилання на id елементів, а не через вкладеність;
– Сумісність із JSON Schema: структура повинна відповідати специфікаціям JSON Schema Draft 2019–09 або новішим, для можливості формальної валідації моделі.

Формалізація вимог до структурного подання UML-моделі у форматі JSON забезпечує створення представлення, яке є інтерпретованим, уніфікованим та яке можна легко перевірити. Така структура придатна до автоматизованої обробки, зокрема, валідації, трансформацій між поданнями, інкрементальної синхронізації та інтеграції в інженерні конвеєри розробки (CI/CD).

Проектування уніфікованої JSON-схеми. Загальна структура уніфікованого JSON-документа, що репрезентує структурне подання UML-моделі, має відповідати принципам формальної коректності, інтерпретованості та розширюваності. Основними елементами є наступні.

Кореневий об’єкт (root) є стартовою точкою документа. Він містить загальні метадані (версію, ідентифікатор, тип моделі), а також вказівки на колекції сутностей. Наприклад:

```
{
  "schemaVersion": "1.0",
  "modelType": "UMLStructure",
  "collections": { "$ref": "#/definitions/collections" }
}
```

Колекції (collections) групують елементи UML-моделі за категоріями: класи, асоціації, пакети тощо. Це дозволяє зберігати логічну організацію моделі. Кожна колекція є масивом об'єктів певного типу:

```
"collections": {
  "classes": [ { "$ref": "#/definitions/Class/U1" }, ... ],
  "packages": [ { "$ref": "#/definitions/Package/P1" } ]
}
```

Механізм посилань ($ref) базується на специфікації JSON Schema та OpenAPI, і забезпечує уніфіковане посилання на внутрішні чи зовнішні елементи моделі. Це дозволяє реалізувати повторне використання, уникати дублювання, та забезпечити чіткість структури.

Опис ключових типів у JSON Schema / OAS. Для формалізації структурного подання UML-моделі у форматі JSON на основі специфікацій JSON Schema та OpenAPI Specification (OAS) визначено ключові типи, які відповідають метасутностям UML: класам, атрибутам, операціям, асоціаціям, пакетам і перелікам (табл. 1). Кожен тип представлено як об'єкт із чітко визначеними полями, обов’язковими атрибутами й валідаційними обмеженнями. Головним типом є UMLClass, що описує сутності домену, їх атрибути (UMLAttribute), операції (UMLOperation), асоціації (UMLAssociation), пакети (UMLPackage) та переліки значень (UMLEnumeration).

Таблиця 1.

Узагальнена таблиця типів у JSON Schema / OAS

Тип	Призначення	Ключові поля	Особливості валідації
UMLClass	Представлення класу UML	id, name, attributes, operations, superClass, isAbstract, stereotypes	Перевірка унікальності id, допустимості типів
UMLAttribute	Опис атрибутів класу	name, type, visibility, defaultValue, multiplicity	Обов’язкові поля, перевірка діапазону значень
UMLOperation	Представлення методів класу	name, parameters, returnType, visibility	Валідація параметрів як масиву об’єктів

Тип	Призначення	Ключові поля	Особливості валідації
UMLAssociation	Визначення асоціації між класами	end1, end2, type, multiplicity, roleName	Обов'язковість двох кінців, правильність типів
UMLPackage	Ієрархічна організація елементів моделі	name, elements, stereotypes	Перевірка коректності імен та вкладеності
UMLEnumeration	Опис перелікових типів	name, literals	literals – перелік рядків із перевіркою унікальності

Усі типи підтримують механізм $ref для модульності та повторного використання, а також розширюються через додаткові поля (additionalProperties). Запропонована структура типів у JSON забезпечує формалізоване подання UML-сутностей, пряме проведення валідації моделей, їх застосування в інструментах моделювання та інтеграцію у процеси CI/CD.

Валідаційні правила подання JSON-схеми. Запропонована JSON-схема підтримує основні типи UML-діаграм із формалізацією відповідних метасутностей та валідаційних обмежень. Основний фокус спрямовано на діаграму класів, де підтримано метасутності Class, Attribute, Operation, Association, Generalization, Package й Enumeration. Діаграма компонентів представлена частково, за допомогою спеціалізованих Component-об'єктів або класів зі стереотипами для забезпечення узгодженості моделі. Діаграма об'єктів поки підтримується опосередковано: екземпляри описуються як об'єкти класів, але повна валідація conformant-інстанціювання поки не реалізована. Діаграма пакетів інтегрована через сутність Package, що дозволяє ієрархічно організовувати структуру моделі.

Запропонований модульний розподіл валідаційних правил має кілька переваг: він дозволяє поступово розширювати модель без ризику порушення стабільних частин, забезпечує цільову перевірку діаграм, знижує складність обчислень та гарантує прозорий зв'язок із UML-специфікацією. Для автоматизованої валідації UML-моделей у JSON-схемі формалізовано структурні обмеження через набір предикатів $\Sigma=\{\sigma_1, \sigma_2,\ldots,\sigma_n\}$, кожен із яких представляє окреме правило коректності моделі.

Згідно з запропонованим підходом, нижче наведено розширене формалізоване подання JSON-схеми UML-діаграми класів із описом ключових сутностей, основних валідаційних вимог і математичної інтерпретації; інваріанти валідації подано в табл. 2.

Визначимо для діаграми класів множину позначень: C – множина класів (UMLClass); A – множина атрибутів (UMLAttribute); O – множина операцій (UMLOperation); R – множина асоціацій (UMLAssociation); G – множина узагальнень (Generalization); ID – множина допустимих ідентифікаторів; означення бінарного відношення узагальнення $Gen \subseteq C \times C$, де $(c, p) \in Gen$ означає «клас c безпосередньо успадковує клас p»; означення множини предків $Ancestors(c) = \{ p \in C \mid (c, p) \in Gen \}$.

Таблиця 2.

Інваріанти валідації класів

№	Назва інваріанта	Формалізація / опис
σ_1	Валідність ідентифікатора класу	$\forall c \in C$: $id(c) \in ID \wedge name(c) \in \Sigma^+$
σ_2	Унікальність імен в межах пакета	$\forall c_1, c_2 \in C$: $package(c_1) = package(c_2) \wedge name(c_1) = name(c_2) \Rightarrow c_1 = c_2$
σ_3	Відсутність циклів узагальнення	$\neg\exists c \in C$: $c \in Ancestors^+(c)$, де $Ancestors(c) = \{p \in C \mid (c,p) \in Gen\}$
σ_4	Унікальність імен атрибутів у класі	$\forall a_1, a_2 \in A$: $class(a_1) = class(a_2) \wedge name(a_1) = name(a_2) \Rightarrow a_1 = a_2$

№	Назва інваріанта	Формалізація / опис
σ_5	Асоціації не є петлями	$\forall r \in R: end_1(r) \neq end_2(r)$
σ_6	Валідність типів атрибутів	$\forall a \in A: type(a) \in CU\{String, Integer, Boolean, Float\}$
σ_7	Унікальність асоціацій між парами	$\forall r_1, r_2 \in R: (end_1(r_1), end_2(r_1)) = (end_1(r_2), end_2(r_2)) \Rightarrow r_1 = r_2$

Наведемо JSON-фрагмент у якості прикладу базової схеми для опису UML-моделі у форматі JSON, зосередженої на двох ключових складових: класах (classes) та асоціаціях (associations). Така структура дозволяє формалізувати діаграму класів з мінімальним, але достатнім набором обмежень для забезпечення її валідації.

```
{
  "type": "object",
  "required": ["classes", "associations"],
  "properties": {
    "classes": {
      "type": "array",
      "items": {
        "type": "object",
        "required": ["id", "name", "attributes"],
        "properties": {
          "id": { "type": "string", "pattern": "^[A-Za-z_][A-Za-z0-
9_]*$" },
          "name": { "type": "string" },
          "attributes": {
            "type": "array",
            "items": {
              "type": "object",
              "required": ["name", "type"],
              "properties": {
                "name": { "type": "string" },
                "type": { "type": "string" }
              }
            },
            "uniqueItems": true
          }
        }
      },
      "uniqueItems": true
    },
    "associations": {
      "type": "array",
      "items": {
        "type": "object",
        "required": ["end1", "end2"],
        "properties": {
          "end1": { "type": "string" },
          "end2": { "type": "string", "not": { "const": { "$data":
"1/end1" } } }
        }
      }
    }
  }
}
```

У межах кореневого об’єкта визначено два обов’язкові поля: масив класів та масив асоціацій. Кожен клас повинен мати унікальний ідентифікатор (id), назву (name) і набір атрибутів (attributes). Ідентифікатор задається рядком, який відповідає регулярному виразу, що унеможливлює використання некоректних символів. Атрибути класу також представлені масивом об’єктів, кожен з яких повинен містити ім’я та тип. Обидва ці поля є обов’язковими, що забезпечує структурну завершеність опису класу.

Крім того, в описі атрибутів та класів використано вимогу uniqueItems, яка забороняє дублювання елементів у межах масиву. Це важливо для збереження унікальності імен класів або атрибутів.

Асоціації описуються як об'єкти з двома кінцями (end1 і end2), які вказують на пов'язані класи. Для запобігання петлям передбачено спеціальне правило, яке забороняє, щоб асоціація зв'язувала клас сам із собою.

Можна вважати, що така схема закладає основу для мінімально достатньої перевірки синтаксичної коректності UML-діаграми класів у JSON-поданні. Водночас, вона може бути розширена додатковими обмеженнями, такими як валідація типів, перевірка існування цільових об'єктів для асоціацій або підтримка успадкування та композиції.

У JSON-схемі, що описує *діаграму пакетів*, основним об'єктом виступає колекція packages. Кожен пакет визначається як окремий JSON-об'єкт, який обов'язково має поля id, name та elements. Формальні вимоги до структури пакета подані у табл. 3.

Для формалізації моделі пакету введемо наступні позначення: P – скінченна множина пакетів; C – множина класів (може бути розширена компонентами, переліками тощо); $E = P \cup C$ – сукупність структурних елементів, які може містити пакет; $id : E \to ID$ – функція, що відображає елемент у його унікальний ідентифікатор; $ID \subseteq \Sigma^+$; $name : E \to \Sigma+$ – функція іменування (людиночитабельності – текстове позначення об'єкта, яке призначене не для машинної обробки, а для сприйняття людиною); $contains \subseteq P \times E$ – бінарне відношення включення (пакет містить елемент).

Для виявлення циклічних залежностей у структурі пакетів використовується транзитивне замикання відношення contains, позначене як $contains^+$, що означає: якщо існує послідовність включень $p_1 \to p_2 \to \cdots \to p_k$, то $(p_1, p_k) \in contains^+$. Це замикання дозволяє перевірити, чи не містить пакет сам себе – прямо або опосередковано.

Таблиця 3.

Інваріанти схеми пакету

№	Назва інваріанта	Формалізація / опис
σ_1	Унікальність ідентифікаторів	$\forall e1,e2 \in E: id(e_1) = id(e_2) \Rightarrow e1 = e2$
σ_2	Унікальність імен у пакеті	$\forall p \in P, \forall e1,e2: (p,e1),(p,e2) \in contains, name(e1) = name(e2) \Rightarrow e1=e2$
σ_3	Ациклічність вкладеності	$\forall p \in P: (p,p) \notin contains^+$
σ_4	Коректність домену відношення	$\forall (p,e) \in contains: p \in P, e \in E$
σ_5	Відсутність дублювання підлеглих елементів	$\forall p \in P, \forall e: (p,e) \in contains \Rightarrow \nexists e' \neq e: (p,e') \in contains \wedge id(e) = id(e')$

Ідентифікатор id задається у вигляді рядка, який повинен відповідати регулярному виразу (наприклад, ^[A-Za-z_][A-Za-z0-9_\\-\\.]*$), що забезпечує як синтаксичну коректність, так і можливість глобальної унікальності в межах моделі (інваріант σ_1).

Поле name призначене для відображення у графічному інтерфейсі й не обов'язково повинне бути унікальним глобально, але має бути унікальним у межах одного пакета (σ_2). Усе вмістиме пакета визначається у полі elements, яке є масивом, що може включати об'єкти типів Class, Package та за потреби – інших сутностей (наприклад, Enumeration, Component). Типізація реалізується через конструкцію oneOf, що дозволяє зберігати гнучкість структури, але гарантує, що всі елементи пакета належать до допустимої множини (σ_4).

Щоб запобігти повторному включенню одного і того ж елемента, до масиву застосовується правило uniqueItems: true, яке в JSON Schema означає перевірку на унікальність (σ_5). Важливо також забезпечити ациклічність ієрархії, тобто, жоден пакет не може включати сам себе, навіть опосередковано. Це перевіряється через обхід графа

включень (наприклад, за допомогою DFS), де контролюється транзитивне замикання відношення contains (σ_3).

Наведемо приклад JSON-фрагменту для пакету UML, який описується як об'єкт, що обов'язково має поля id, name та elements.

```
{
  "$id": "https://example.org/uml/package-schema.json",
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "title": "UML Package",
  "type": "object",
  "required": ["id", "name", "elements"],
  "properties": {
    "id": {                          // σ1: Унікальні id
      "type": "string",
      "pattern": "^[A-Za-z_][A-Za-z0-9_\\-\\.]*$"
    },
    "name": { "type": "string" }, // Людиночитабельне ім'я (σ2)
    "elements": {
      "type": "array",
      "items": { "oneOf": [
        { "$ref": "class-schema.json" },
        { "$ref": "package-schema.json" }
      ]},
      "uniqueItems": true          // σ5: уникнення структур-дублікатів
    },
    "stereotype": { "type": "string" },
    "tags": {
      "type": "object",
      "additionalProperties": {
        "type": ["string", "number", "boolean"]
      }
    }
  }
}
```

Ідентифікатор задається рядком, що відповідає ^[A-Za-z_][A-Za-z0-9_-.]*$; схема відсікає некоректні значення, а зовнішній валідатор контролює глобальну унікальність. Поле name задає читабельну назву пакета, а окреме правило стежить, аби всередині пакета не було дубльованих назв.

Масив elements через oneOf приймає лише класи та вкладені пакети, тим самим підтримуючи замкненість домену; uniqueItems: true блокує дослівні дублікати, а повторні id перехоплює зовнішній скрипт.

Щоби запобігти рекурсивному включенню пакетів, у CI/CD-конвеєрі запускається обхід графа «пакет містить пакет»; якщо на транзитивному шляху виявляється вузол, що повертається до самого себе, така модель відхиляється.

Поля stereotype та tags є необов'язковими: перше фіксує роль або категорію пакета, друге дозволяє додавати довільні ключ-значення доменної інформації без ризику порушити основні інваріанти.

Уніфіковане подання діаграми компонентів у форматі JSON слугує формальним способом опису архітектурної композиції системи на високому рівні абстракції. Основною метасутністю є Component – логічна одиниця розгортання, яка може мати інтерфейси, залежності, порти, а також пов'язуватись із класами або іншими компонентами.

У JSON-схемі об'єкти компонентів описуються масивом components, де кожен елемент є окремим об'єктом із унікальним id, ім'ям name, та (за потреби) специфікацією interfaces та dependencies. Також передбачена можливість вкладення компонентів, наприклад, для мікросервісної архітектури.

Визначимо загальну структуру JSON-схеми, що описує діаграму компонентів: components – масив компонентів; кожен компонент – об'єкт з полями id, name, interfaces,

dependencies; interfaces – перелік назв або ідентифікаторів точок взаємодії; dependencies – ідентифікатори інших компонентів, від яких залежить даний компонент.

Для забезпечення валідації структура включає обов'язковість ключових полів, верифікацію ідентифікаторів за регулярними виразами, перевірку унікальності та відсутності циклічних залежностей (табл. 4).

Для формалізації моделі компонентів введемо наступні позначення: C – множина всіх компонентів моделі; $id : C \to ID \subset \Sigma^+$ – функція, що відображає компонент у його унікальний ідентифікатор; $name : C \to \Sigma^+$ – функція людиночитабельного імені; $dep \subseteq C \times C$ – бінарне відношення залежностей між компонентами; $iface : C \to 2^{\Sigma^+}$ – функція, яка для кожного компонента визначає множину його інтерфейсів.

Транзитивне замикання dep^+ використовується для визначення наявності непрямих залежностей між компонентами. Воно визначається як об'єднання всіх можливих послідовних застосувань відношення залежності (композиція на себе k разів, $k \geq 1$).

Таблиця 4.

Інваріанти валідації компонентів

№	Назва інваріанта	Формалізація / опис
σ_1	Унікальні ідентифікатори	$\forall c1, c2 \in C : id(c1) = id(c2) \Rightarrow c1 = c2$
σ_2	Унікальні імена	$\forall c1, c2 \in C : name(c1) = name(c2) \Rightarrow c1 = c2$
σ_3	Ациклічність залежностей	$\forall c \in C : c \notin dep^+(c)$
σ_4	Валідність посилань у залежностях	$\forall (c1,c2) \in dep : c1 \in C \wedge c2 \in C$
σ_5	Унікальність інтерфейсів у компоненті	$\forall c \in C$:iface(c) – множина без повторень

Ці обмеження виконуються як частково засобами JSON Schema (σ_1, σ_2, σ_5), так і зовнішніми валідаторами або додатковими пост-обробниками (σ_3, σ_4).

Наведемо приклад опису компонентів у JSON:

```
{
    "components": [{
            "id": "AuthService",
            "name": "Authentication",
            "interfaces": ["login", "logout", "tokenValidate"],
            "dependencies": ["UserDB"]
        }, {
            "id": "UserDB",
            "name": "User Database",
            "interfaces": ["readUser", "writeUser"]
        }
    ]
}
```

У цьому прикладі описано два компоненти: AuthService і UserDB. Кожен має унікальний id та читабельну назву name. Компонент AuthService реалізує інтерфейси login, logout, tokenValidate і залежить від UserDB. Всі поля відповідають правилам: id є строковим значенням, що задовольняє регулярний вираз, масиви інтерфейсів не містять повторень, і залежність не породжує циклів. Структура відповідає інваріантам σ_1–σ_5 і готова до автоматизованої перевірки.

У випадку діаграми об'єктів основна мета JSON-схеми полягає в представленні конкретних екземплярів класів, що відповідають об'єктам моделі. Така схема є доповненням до діаграми класів і забезпечує семантично точне відображення стану системи на певному етапі її виконання або проектування.

JSON-схема для діаграми об'єктів містить масив objects, кожен елемент якого описує окремий екземпляр класу. Об'єкт обов'язково включає поля: id – унікальний ідентифікатор екземпляра; class – посилання на відповідний клас (за id класу з UML-моделі); slotValues – об'єкт, що є словником значень атрибутів, де ключ – це ім'я

атрибута, а значення – значення цього атрибута; links – необов'язковий масив, що задає зв'язки з іншими об'єктами (відповідає асоціаціям).

Крім того, для збереження узгодженості моделі застосовуються обмеження: кожен class має бути валідним посиланням на визначений у діаграмі класів тип; усі slotValues повинні відповідати типам атрибутів, що описані в класі; об'єкти, що беруть участь у links, мають бути оголошені у межах цієї ж діаграми; значення id кожного об'єкта має бути унікальним у межах всієї діаграми.

Для формалізації моделі об'єктів введемо наступні позначення: O – множина об'єктів; C – множина класів із UML-діаграми; A(c) – множина атрибутів класу c∈C; V – множина значень; $id : O \to ID \subseteq \Sigma^+$ – функція, що задає унікальний ідентифікатор об'єкта; $class : O \to C$; $value : O \times A(class(o)) \to V$ – функція, що визначає значення кожного атрибута; $link \subseteq O \times O$ – бінарне відношення між об'єктами (відповідає асоціаціям).

Система обмежень формалізована у таблиці інваріантів, яка забезпечує унікальність ідентифікаторів, відповідність атрибутів класам, коректність типів значень та відсутність посилань на неіснуючі об'єкти (табл. 5).

Таблиця 5.

Інваріанти валідації об'єктів

№	Назва інваріанта	Формалізація / опис
1	Унікальність id	$\forall o_1, o_2 \in O: id(o_1) = id(o_2) \Rightarrow o_1 = o_2$
2	Належність класу	$\forall o \in O: class(o) \in C$
3	Наявність значень для атрибутів	$\forall o \in O, \forall a \in A(class(o)): \exists v \in V: value(o, a) = v$
4	Коректність посилань	$\forall (o_1, o_2) \in link: o_1 \in O \land o_2 \in O$
5	Відповідність типів значень	$\forall o \in O, \forall a \in A(class(o)): type(value(o, a)) \in allowed\_types(a)$

Наведемо JSON-приклад об'єктів obj1 і obj2, що є екземплярами класу Person, який повинен бути визначений у діаграмі класів з атрибутами name та age.

```
{
 "objects":[
 {
       "id":"obj1",
       "class":"Person",
       "slotValues":{"name":"Alice","age":30},
       "links":["obj2"]
 },
 {
       "id":"obj2",
       "class":"Person",
       "slotValues":{"name":"Bob","age":35},
       "links":["obj1"]
 }]
}
```

Значення атрибутів відповідають очікуваним типам (string, integer). Поле links відображає взаємозв'язок між об'єктами, що в UML відповідає бінарній асоціації між екземплярами класу.

Такий підхід забезпечує формальну відповідність між класовою і об'єктною діаграмами, дозволяючи проводити типову й структурну валідацію даних, підтримувати тестування моделей та їх верифікацію в інструментах UML-моделювання.

Для уніфікованої реалізації перевірки відповідності UML-моделей у JSON-форматі нижче подано узагальнену таблицю валідаційних правил, що реалізуються на рівні JSON Schema або зовнішнього валідатора (табл. 6).

Таблиця 6.

Узагальнення валідаційних правил

№	Група правил	Формальна/схемна реалізація	Призначення
R-1	Обов'язкові поля	required у кореневих та вкладених об'єктах minItems: 1 для масивів classes, packages, components, objects	Гарантія мінімально необхідної інформації; порожній масив класів або відсутність id у елемента – критична помилка
R-2	Універсальний ідентифікатор	json pattern: "^[A-Za-z_][A-Za-z0-9_\\-\\.]*$" propertyNames для словників (напр. тегів)	Єдина регулярна форма id для всіх типів елементів. Літери, цифри, підкреслення, дефіс, крапка; починатися має з літери/_
R-3	Перелік допустимих примітивних типів	json { "enum": ["Boolean","Integer","Real","String","Date","Time"] }	Вирівнює типи атрибутів/параметрів з UML 2.5; запобігає довільним або помилковим позначенням (int, str, …)
R-4	Унікальність у масивах	uniqueItems: true та/або користувацький ключ uniqueItemProperties: ["id"] (підтримується бібліотекою Ajv)	Забороняє дублювати елементи в межах колекції (класи в пакеті, атрибути в класі, інтерфейси в компоненті)
R-5	Крос-посилання ($ref/$data)	перевірка типу: "$ref": "#/$defs/ClassId" короткий $data-rule: "not": { "const": { "$data": "1/clientId" } }	Виконує семантичну узгодженість: кожне class, supplierId, superClassId тощо має співпадати з реально оголошеним id
R-6	Валідація імен (людиночитабельних)	json pattern: "^[A-Z][A-Za-z0-9_]*$" (допустимі також локалізовані літери)	Уникає порожніх/невалідних назв; полегшує читабельність у графічних редакторах
R-7	Ациклічність ієрархій	зовнішній алгоритм обходу графа DFS / топологічне сортування для contains, generalization, dependency	Логічна перевірка, яку складно записати чистими ключами JSON Schema; виконується у скрипті CI/CD
R-8	Контроль типів значень у діаграмі об'єктів	oneOf / if…then…else на основі $data : якщо атрибут оголошено як "Integer" → type: "integer", якщо "String" → type: "string"	Забезпечує, щоб екземпляр об'єкта реально дотримувався типу, визначеного в класі
R-9	Розширюваність (стереотипи, теги)	поле tags описується як "additionalProperties": { "type": ["string","number","boolean"] }	Дозволяє довільно додавати метадані без зміни базової схеми; зберігає forward-compatibility

Результати та обговорення. Аналіз перевірки UML-моделей показав, що близько 70 % інваріантів Σ реалізуються засобами JSON Schema (required, enum, pattern, uniqueItems, if…then…else), що охоплює перевірку синтаксису, структури й типізації. Проте для 30 % складніших залежностей (ациклічність відношень успадкування та включення, транзитивну узгодженість зв'язків, цілісність міжоб'єктних посилань, унікальність імен у вкладених просторах) стандартної декларативної виразності недостатньо.

Для таких перевірок застосовується розширена валідація через зовнішні процедури (after-hooks) або CI/CD-механізми, які дозволяють інтерпретувати модель як граф і виконати обхід (наприклад, DFS чи топологічне сортування) для підтвердження інваріантів, що потребують глобального або транзитивного контексту. У межах цих процедур: JSON-файл з моделлю парситься у вигляді графу; окремі модулі перевіряють інваріанти σ_i, які потребують глобального контексту; результати перевірок повертаються у вигляді узагальненого звіту або блокують конвеєр, якщо виявлено структурні помилки.

Висновки. У роботі сформульовано та реалізовано формалізоване структурне подання UML-метамоделі у форматі JSON, що охоплює основні метасутності (класи, атрибути, асоціації, узагальнення, пакети, компоненти, об'єкти) та відповідні відношення між ними. Запропоноване подання базується на системі JSON Schema з урахуванням формальних інваріантів структурної цілісності, які задають правила іменування, вкладеності, типізації та відсутності циклів.

Практична апробація моделі показала, що більшість інваріантів може бути реалізована стандартними засобами JSON Schema, однак низка критичних перевірок (зокрема, транзитивні й міжсхемні залежності) потребує зовнішніх процедур контролю. Це зумовлює необхідність інтеграції розширених валідаційних механізмів у середовища CI/CD або спеціалізовані інструменти перевірки UML-моделей.

Запропоноване подання метамоделі створює підґрунтя для подальших досліджень у напрямках: автоматизованої верифікації цілісності структурних моделей; трансформації UML-діаграм у мови програмування або формати XMI; побудови відкритого інструментарію для аналізу й редагування UML-моделей у JSON форматі.

Список літератури

1. Komleva N. O., Nikitchenko M. I. Method for incremental control of consistency between structural and behavioral views of software architecture. Applied Aspects of Information Technology. 2025. Vol. 8, No. 2. (в друці) DOI: https://doi.org/10.15276/aait.08.2025.6.
2. Nikitenko M. I. Analysis of formats for storing UML models in modern CASE tools. Technology and Society: Interaction, Influence, Transformation: Proceedings of the IV International Scientific Conference (20 June 2025, Chernihiv, Ukraine). 2025. P. 208–212. DOI: https://doi.org/10.62731/mcnd-20.06.2025.
3. Clarisó R., González C. A., Cabot J. Incremental verification of UML/OCL models using logical abstraction. Journal of Object Technology. 2020. Vol. 19, No. 3. P. 1–20. DOI: https://doi.org/10.5381/jot.2020.19.3.a7.
4. Shaikh A., Wiil U. K. Overview of slicing and feedback techniques for efficient verification of UML/OCL class diagrams. IEEE Access. 2018. Vol. 6. P. 23864–23882. DOI: https://doi.org/10.1109/ACCESS.2018.2797695.
5. Wright A., Andrews H., Hutton B. JSON Schema: A media type for describing JSON documents [Електронний ресурс]. 2019. URL: https://json-schema.org/draft/2019-09/json-schema-core.html.
6. Wright A., Andrews H., Hutton B. JSON Schema Validation: A vocabulary for structural validation of JSON [Електронний ресурс]. 2020. URL: https://json-schema.org/draft/2020-12/json-schema-validation.html.
7. Portele C., Echterhoff J., et al. Best practice for OGC – UML to JSON encoding rules [Електронний ресурс]. 2020. URL: http://www.opengis.net/doc/BP/uml-to-json-encoding/1.0.
8. Attouche L., Baazizi M.-A., Colazzo D., Ghelli G. Validation of modern JSON Schema: formalization and complexity. arXiv preprint. 2023. DOI: https://doi.org/10.48550/arXiv.2307.10034.
9. Attouche L., Baazizi M.-A., Colazzo D., Ulliana F. Elimination of annotation dependencies in validation for modern JSON Schema. arXiv preprint. 2025. arXiv: 2503.11288. URL: https://arxiv.org/abs/2503.11288.

10. Khan A. H., Porres I. Consistency of UML class, object and statechart diagrams using ontology reasoners. arXiv preprint. 2022. arXiv: 2205.11177. URL: https://arxiv.org/abs/2205.11177.
11. Shaikh A., Khan A. H., Wiil U. S. More than two decades of research on verification of UML class models: A systematic literature review. IEEE Access. 2021. Vol. 9. P. 128266–128295. DOI: https://doi.org/10.1109/ACCESS.2021.3121222.
12. SOM-Research. JSONSchema to UML [Електронний ресурс]. 2022. URL: https://github.com/SOM-Research/jsonschema-to-uml.
13. StarUML Documentation. StarUML model file structure [Електронний ресурс]. 2024. URL: https://docs.staruml.io/user-guide/managing-project.
14. Modelio Documentation. Modelio JSON export format [Електронний ресурс]. 2023. URL: https://www.modelio.org/index.htm.
15. Hafeez A., Musavi H. A., Rehman A.-u. Ontology-Based Transformation and Verification of UML Class Model. The International Arab Journal of Information Technology. 2020. Vol. 17, No. 5. P. 758–768. DOI: https://doi.org/ 10.34028/iajit/17/5/9.
16. Di Ruscio D., Franzago M., Malavolta I., Muccini H. Envisioning the Future of Collaborative Model-Driven Software Engineering. 2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C), Buenos Aires, Argentina. 2017. P. 173–175. DOI: https://doi.org/10.1109/ICSE-C.2017.143.
17. Döller V. Formalizing the four-layer metamodeling stack with MetaMorph: potential and benefits. Software and Systems Modeling. 2022. Vol. 21. P. 1411–1435. DOI: https://doi.org/10.1007/s10270-022-00986-2.

STRUCTURAL VIERW OF THE UML METAMODEL IN JSON FORMAT

N.O. Komleva, M.I. Nikitchenko

Odessa Polytechnic National University
1, Shevchenko Ave., Odesa, 65044, Ukraine
Emails: komleva@op.edu.ua, maksym.nikitchenko@gmail.com

The paper considers the problem of formal representation of UML (Unified Modeling Language) models in an open format suitable for verification, transformation, and integration into software development engineering processes. The conceptual basis of the study is the UML metamodel, which includes two interrelated views: structural (defines objects and model relationships) and behavioral (describes activities, states, and sequences). This metamodel serves as the basis for incremental control of software architecture consistency. In this work, special attention is paid to the structural view of the UML metamodel, which is the basis for representing the system architecture in the form of interrelated object types. For this view, a formalized representation of key UML meta-entities, such as classes, attributes, associations, packages, components, and objects, is implemented in JSON (JavaScript Object Notation) format. The chosen view is based on the application of the JSON Schema standard, which allows for a clear description of valid structures, data types, named properties, and validation rules. As a result, a complete set of types and relationships has been constructed, which defines the allowed configuration of the UML model as a structured set of objects with clearly defined relationships between them. To support the syntactic and semantic integrity of this model, a system of formalized invariants has been formulated, which fixes critical constraints, in particular, requirements for identifiers, correct nesting of elements, strict typing of attributes, and the absence of cyclic dependencies in composite and inherited structures. This lays the foundation for a unified representation of structural models and further automated verification of their correctness. Some of the rules are implemented by the JSON schema itself, while more complex ones are implemented through external verification mechanisms. The proposed approach ensures a clear correspondence between abstract UML objects and their representation in an open format, which, in the long term, opens up the possibility of unified integration with other modeling tools, CI/CD, and code generation tools. The research focuses on formalism, transparency, and consistency with the UML specification, which are necessary to support incremental consistency control while ensuring compatibility between the components of the metamodel.
Keywords: modeling, UML, metamodel, JSON Schema, validation, invariants.

DOI 10.15276/imms.v15.no2.218
УДК 004.056.53

Informatics and Mathematical Methods in Simulation
Vol.15 (2025), No. 2, pp. 218-228

МЕТОДИ ГЕНЕРАЦІЇ ТА АНАЛІЗУ СТІЙКОСТІ ПАРОЛІВ ІЗ УРАХУВАННЯМ ЛЕКСИЧНИХ, СТРУКТУРНИХ ТА ЕВРИСТИЧНИХ ОЗНАК

Т.Ю. Кришталь, О.В. Троянський, Н.І. Кушніренко

Національний університет «Одеська політехніка»
1, Шевченка пр., м.Одеса, 65044, Україна
Emails: kushnirenko@op.edu.ua, o.v.troyanskiy@op.edu.ua

У статті розглянуто розроблені методи генерації та аналізу стійкості паролів із урахуванням лексичних, структурних та евристичних ознак, що були реалізовані в рамках розробки програмного продукту KrystalLock. Основною метою роботи стало створення ефективних методів, що забезпечують як генерацію, так і перевірку паролів на предмет відповідності сучасним вимогам безпеки. У межах реалізованого підходу запропоновано три методи генерації паролів: випадковий, мнемонічний та комбінований. Мнемонічна генерація ґрунтується на створенні паролів із фраз, які легко запам'ятовуються користувачем, з подальшим застосуванням вибіркових leet-перетворень для підвищення складності. Всі методи підтримують налаштування параметрів, зокрема довжину пароля, використання окремих категорій символів (великі/малі літери, цифри, спецсимволи), а також контроль над змістовними шаблонами. У частині аналізу стійкості паролів реалізовано багаторівневий метод, що поєднує класичні критерії оцінки складності з сучасними підходами до машинного аналізу. Зокрема, здійснюється оцінка довжини, символьного складу, виявлення шаблонів (повторення, послідовності, розповсюджені клавіатурні комбінації), перевірка на наявність у відомих злитих базах паролів, а також лексичний та семантичний аналіз. Окрему увагу приділено машинному підходу до виявлення слабких паролів, що побудований на моделі двонаправленої довготривалої пам'яті (BiLSTM). Модель навчена на датасеті паролів з які містять різноманітні імена та слова та демонструє точність понад 92% на тестових вибірках, використовуючи метрики F1, Precision та Recall. Розроблений застосунок реалізований у середовищі Python з використанням бібліотек TensorFlow, NumPy, secrets та інших. Проведено тестування розроблених методів, включаючи як функціональні, так і навантажувальні сценарії. Оцінено відповідність результатів аналізу очікуваним стандартам кібербезпеки (NIST, OWASP, ISO/IEC 27001) та підтверджено ефективність підходу порівняно з існуючими аналогами. Результати роботи можуть бути використані для підвищення рівня захисту облікових записів у персональних та корпоративних середовищах, а також для впровадження у системи керування паролями. Застосунок може стати основою для подальших досліджень і вдосконалення в галузі адаптивної аутентифікації, зокрема за рахунок інтеграції поведінкового аналізу або біометричних факторів. Комплексність підходу, використання моделей глибокого навчання та можливість персоналізації роблять розроблений продукт цінним внеском у сферу практичної кібербезпеки.

Ключові слова: стійкість паролів, генерація паролів, машинне навчання, BiLSTM, мнемонічна генерація, семантичний аналіз, кібербезпека.

Вступ. У сучасну епоху цифрової трансформації питання безпеки облікових даних набуває особливої актуальності. Інформаційні системи дедалі частіше стають мішенню зловмисників, а успішна компрометація облікових записів може призвести до масштабних витоків персональної, фінансової або корпоративної інформації. Попри розвиток технологій багатофакторної автентифікації та впровадження біометричних рішень, текстові паролі залишаються найпоширенішим механізмом захисту — простим, універсальним і зручним, але водночас надзвичайно вразливим.

Статистичні звіти у сфері кібербезпеки свідчать, що понад 80% інцидентів з несанкціонованим доступом до систем пов'язані з використанням ненадійних або повторно вживаних паролів [1]. Значна частка користувачів і досі створює прості комбінації типу «123456», «qwerty» або використовує особисту інформацію, таку як дати

народження, імена родичів або домашніх тварин. Ці шаблони легко виявляються за допомогою словникових атак або методів соціальної інженерії. Нерідко паролі не оновлюються роками, а один і той самий пароль застосовується одночасно для кількох сервісів, що створює ефект «доміно» у разі компрометації лише одного з них.

Сучасні вимоги до паролів визначають не лише їх довжину та символьний склад, а й здатність протистояти автоматизованому підбору, прогнозованості та перевіркам на наявність у скомпрометованих базах. Національний інститут стандартів і технологій США (NIST) у публікації SP 800-63B, а також міжнародний стандарт ISO/IEC 27001 рекомендують створення унікальних, складних і довгих паролів, зокрема за допомогою спеціалізованих генераторів [2]. Проте, навіть дотримання формальних критеріїв не завжди гарантує фактичну стійкість пароля — наприклад, «Password123!» технічно відповідає більшості вимог, але на практиці є вразливим через поширеність структури.

Класичні системи перевірки паролів переважно базуються на жорстких правилах (rule-based), які оцінюють пароль за простими метриками: наявність великих літер, цифр, символів тощо. Однак ці підходи не враховують семантичного змісту, контексту чи ймовірності повторення шаблонів. У відповідь на ці обмеження почали впроваджуватися інтелектуальні методи — моделі машинного навчання, здатні аналізувати структуру паролів, визначати латентні шаблони вразливості, проводити класифікацію та прогнозування потенційної стійкості пароля. Особливо перспективним напрямом є використання нейронних мереж, зокрема архітектури BiLSTM (Bidirectional Long Short-Term Memory), що дозволяє враховувати послідовну природу символьних ланцюжків [3].

Окрему увагу у сфері захисту облікових даних починає привертати проблема «користувацького комфорту» — складний пароль часто складно запам’ятати, а прості — зазвичай слабкі. Для подолання цього протиріччя з’являються альтернативні методи генерації — зокрема мнемонічні алгоритми, які дозволяють створювати паролі на основі фраз або структур, зручних для користувача. Наприклад, фраза «I love going to the gym at 7am!» може бути перетворена у пароль типу «I1g!tTg@7AM!». Це забезпечує баланс між зручністю та стійкістю, однак потребує додаткової обробки для уникнення вразливостей до шаблонів.

У відповідь на вищезгадані виклики в рамках цієї роботи розроблено комплексний застосунок, що об’єднує модуль генерації паролів із можливістю мнемонічної та випадкової побудови, модуль глибокого аналізу з використанням моделей машинного навчання, а також інтерфейс, що забезпечує доступну інтерпретацію результатів перевірки. Такий підхід дозволяє поєднати гнучкість, наукову обґрунтованість і практичну корисність, формуючи новий етап у боротьбі за безпеку цифрових ідентичностей.

Огляд літератури. Сучасні дослідження у сфері безпеки систем автентифікації засвідчують, що текстові паролі залишаються основним інструментом захисту облікових записів, попри стрімкий розвиток біометричних технологій та багатофакторної автентифікації [4]. Паролі розглядаються як ключовий засіб первинної перевірки особистості користувача, однак через низьку якість їх створення з боку самих користувачів, вони часто стають основною причиною компрометації облікових даних. Для підвищення рівня стійкості паролів міжнародні організації, такі як Національний інститут стандартів і технології США (NIST) та Міжнародна організація зі стандартизації (ISO), розробили відповідні вимоги до їх формування. У документах NIST SP 800-63B [5] та ISO/IEC 27001 визначено ключові рекомендації: мінімальна довжина від 10 до 16 символів, використання різних категорій символів, унікальність, а також перевірка на відповідність базам зламаних паролів [2, 6].

Оцінка надійності пароля передбачає використання кількох критеріїв: ентропії, символьного складу, структурних шаблонів та стійкості до словникових атак [4, 7]. Наприклад, пароль «SmartP@ss123», який містить символи з різних категорій, формально відповідає основним вимогам, однак може бути легко вгаданий через

поширеність шаблону. Такі комбінації часто піддаються модифікованим словниковим атакам, які використовують типові патерни трансформації, зокрема заміну літер на цифри чи спецсимволи [8, 9]. Крім того, дослідники відзначають високий рівень повторного використання паролів. За результатами масштабного аналізу понад 60 млн паролів, проведеного Wang та інш., близько 38% користувачів застосовували однакові комбінації для різних сервісів [1].

Для підвищення стійкості паролів запропоновано кілька альтернатив традиційному ручному створенню. Одним із найбільш популярних є мнемонічний підхід, що базується на трансформації фраз або слоганів у складні комбінації. Наприклад, фраза «Моя собака народилась у 2020 році» може бути перетворена на «Msnu2020r» (транслітеровано). Цей метод забезпечує високу запам'ятовуваність, але, як зазначається в роботах [7, 10], не гарантує стійкість до евристичного зламу або семантичного аналізу. Більш ефективним вважається поєднання мнемоніки з вибірковим leet-перетворенням — заміною символів на нестандартні аналоги («а» → «@», «s» → «$», тощо), що підвищує ентропію пароля та ускладнює його прогнозування [7].

Сучасні аналітичні системи, зокрема VpnMentor та PeriTools, реалізують багаторівневу оцінку складності паролів. Вони враховують структуру, довжину, наявність словникових елементів та відповідність скомпрометованим базам [11, 12]. Окрім візуальних індикаторів складності, що використовуються у веб інтерфейсах, розробляються і більш глибокі алгоритмічні моделі аналізу. Наприклад, нейронні мережі типу BiLSTM, згідно з дослідженнями [3], дозволяють точно класифікувати паролі за категоріями стійкості, обробляючи їх як послідовності текстових даних. Ці моделі ефективно виявляють латентні семантичні шаблони, що не піддаються виявленню за допомогою класичних методів.

У сфері генерації криптографічно стійких паролів стандартом де-факто стала мова програмування Python у поєднанні з модулем secrets, який забезпечує генерацію випадкових значень з криптографічною ентропією [3, 13]. Бібліотеки TensorFlow, NumPy та sklearn використовуються для реалізації та тренування моделей машинного навчання, включно з класифікаторами на основі випадкових лісів, логістичної регресії та глибинних нейронних мереж [13-15]. Такий підхід дозволяє будувати комплексні системи оцінки, що враховують як поверхневі характеристики (довжина, символи), так і лінгвістичні, структурні та навіть поведінкові особливості користувачів при створенні паролів.

Таким чином, сучасна література демонструє значний інтерес до створення інтелектуальних систем для аналізу й генерації паролів. Поєднання лексичного, структурного та евристичного аналізу із методами глибокого навчання, зокрема BiLSTM, формує новий стандарт забезпечення стійкості автентифікаційних даних у цифрових екосистемах. Саме така багатофакторна оцінка, доповнена можливістю персоналізованої генерації, стала базою для побудови програмного продукту, представлення якого наведено у подальших розділах.

Мета роботи. Метою дослідження є розробка нових методів генерації та аналізу стійкості паролів, що поєднують структурні, лексичні та евристичні ознаки, з урахуванням сучасних вимог до інформаційної безпеки. Запропоновані методи орієнтовані на підвищення ефективності виявлення слабких паролів та формування криптографічно надійних комбінацій за рахунок використання класичних підходів у поєднанні з моделями глибокого навчання.

Для реалізації поставленої мети було визначено такі основні задачі:

– здійснити аналіз сучасних підходів до генерації та перевірки паролів, визначити їхні переваги і недоліки;

– сформувати методи генерації паролів, що враховують не лише ентропійні параметри, а й зручність запам'ятовування (мнемонічні особливості) та структурну складність;

– розробити підхід до комбінованої генерації, який поєднує мнемонічні та криптографічно стійкі елементи;

– створити метод багаторівневого аналізу стійкості паролів з урахуванням лексичних, шаблонних і семантичних характеристик;

– реалізувати розроблені методи в програмного застосунку та протестувати їх ефективність.

Основна частина. У межах роботи було розроблено програмне забезпечення KrystalLock — інтерактивний застосунок, призначений для генерації надійних паролів і оцінки їхньої стійкості. Система поєднує класичні, мнемонічні та нейронні підходи, забезпечуючи комплексний аналіз як із погляду структурної складності, так і з урахуванням лексичних та семантичних ризиків. Основна ідея полягає не лише в автоматизації створення паролів, а у формуванні критичного мислення в користувача щодо надійності автентифікаційної інформації. Програмне забезпечення реалізує три незалежні, але взаємопов'язані методи генерації паролів: випадковий, мнемонічний та комбінований. Одним з запропонованих методів в рамках реалізованого програмного забезпечення KrystalLock є застосування генератора паролів, що працює на основі криптографічно стійкого випадкового вибору символів. Такий підхід гарантує високий рівень ентропії за рахунок непередбачуваності кожного елемента послідовності. У межах реалізації використовувалась стандартна бібліотека secrets мови програмування Python, яка забезпечує генерацію псевдовипадкових чисел з використанням криптографічних механізмів, на відміну від генераторів типу random, що є статистично, але не криптографічно безпечними.

Процес генерації пароля включає декілька етапів. На початковому рівні потрібно вказати необхідні параметри: бажану довжину пароля, а також категорії символів, що повинні входити до складу комбінації. Підтримуються чотири основні класи: великі латинські літери (A–Z), малі латинські літери (a–z), цифри (0–9) та спеціальні символи (наприклад: !, @, #, $, %, & тощо). Залежно від вибору, формується набір допустимих символів, з якого далі здійснюється вибір кожного елементу пароля з використанням secrets.choice().

Формально алгоритм роботи методу генерації можна подати у вигляді:

$$P = \text{join}(\text{choice}(S_i))_{i=1}^{n},\ S_i \subseteq \{A-Z, a-z, 0-9, \text{symbols}\},\ n = \text{length} \qquad (1)$$

де P — згенерований пароль, S_i — набір символів, доступних для вибору на *i*-й позиції, n — загальна довжина пароля.

Важливо зазначити, що реалізація враховує правило мінімального представництва, тобто гарантує, що в результаті будуть присутні символи з кожної вибраної категорії. Це реалізовано через обов'язкове включення принаймні одного символу з кожного класу, після чого решта позицій заповнюється випадковим чином із повного об'єднаного алфавіту. Після створення всі символи перемішуються, аби уникнути передбачуваних шаблонів (наприклад, коли символи певного типу завжди розміщені на початку). Паролі, створені за цією методикою, демонструють високі показники ентропії, оскільки всі символи рівноймовірні й незалежні між собою. Такий підхід значно ускладнює реалізацію атак типу brute-force або dictionary-based.

Набагато цікавішою з точки зору інтелектуальної складності є реалізація мнемонічного методу генерації, для якої в роботі було розроблено п'ять чітких правил трансформації вхідної фрази, що адаптуються до її довжини. Кожне правило передбачає специфічну логіку вибору слів та їх перетворення із застосуванням вибіркового leet-перетворення.

Перше правило застосовується у випадку, коли користувач вводить лише одне слово. У цьому випадку програма виконує leet-заміну символів відповідно до табл. 1. однак не кожна заміна відбувається зі 100% вірогідністю. Наприклад, слово nature може бути перетворене у n@7ur3, однак при іншому запуску — у na7urE.

Таблиця 1

Найпопулярніші leet-заміни літер

Літера	**LEET-заміна**
A	@
B	8
E	3
H	#
I	1
L	£
O	0
S	$
T	7
Z	2

Друге правило стосується коротких фраз (2–3 слова). У цьому випадку одне зі слів вибирається випадково та трансформується leet-замінником, інші — скорочуються або зводяться до першої літери. Наприклад, фраза nature is beautiful може бути перетворена у ntr1$b, де “ntr” — скорочення від “nature”, “1$” — трансформація “is”, а “b” — перша літера слова “beautiful”.

Третє правило застосовується для фраз середньої довжини (4–5 слів). У цьому випадку вже два слова випадково піддаються leet-заміні, інші знову ж або скорочуються, або абревіатуризуються. Наприклад, nature is so beautiful now трансформується у nTri$0Be@u71FU£n.

Четверте правило охоплює фрази довжиною 6–9 слів. Воно подібне до попереднього, однак трансформації піддаються вже три випадково обрані слова. Приклад перетворення: nature is so beautiful now isn’t it → Nis83@u7IFu£now1$N7i`.

П’яте правило стосується фраз, які містять більше 9 слів. У цьому випадку реалізується агресивна абревіатуризація: 90% слів замінюються лише першими літерами, а 10% — скорочуються. Наприклад, фраза nature is so beautiful now, isn’t it especially in spring bloom може бути представлена у вигляді Nisbniiepclyisb.

Всі трансформації реалізуються із використанням інтелектуального механізму вибору leet-замін — система надає перевагу символам, які легко вводити користувачеві, не порушуючи баланс між зручністю і стійкістю. На відміну від примітивного повного заміщення, вибірковість знижує передбачуваність і захищає від атак на основі leet-словників.

Окремо слід розглянути комбінований метод, який є синтезом мнемонічної генерації, описаної вище, та криптографічної компоненти. До згенерованого мнемонічного пароля додається випадкова послідовність символів — на початку, в кінці або з обох сторін. Місце додавання та довжина фрагменту варіюється, забезпечуючи додаткову ентропію та розрив шаблону. Наприклад: ntr1$b → 9!ntr1$b або 9!ntr1$b@W7.

Базова оцінка стійкості паролів в системі реалізується за допомогою двох підходів — класичного (комплексного) і нейронного (машинного). У першому випадку використовується формула базової ентропії:

$$H = L \log_2 R \tag{2}$$

де L — довжина пароля, R— потужність алфавіту. Наприклад, для 10-символьного пароля з 62 символами (a-z, A-Z, 0-9) отримаємо $H = 10 \log_2(62) \approx 59.54$ біт. Проте ця формула справедлива лише для рівномірно випадкових паролів. Для реальних, семантично осмислених комбінацій застосовується модифікована формула Шеннона:

$$H = \sum_{i=1}^{n} p_i \log_2 p_i \tag{3}$$

де p_i — імовірність появи i-го символу. Таке оцінювання дозволяє враховувати повторюваність символів і зменшувати ентропію за наявності шаблонів (наприклад: 12345678, qwerty, admin).

Машинний аналіз стійкості реалізовано за допомогою двонаправленої нейронної мережі BiLSTM (Bidirectional Long Short-Term Memory). Модель навчена на датасеті розміром 513 853 паролів, транслітерованих українських та англійських слів та імен. Архітектура мережі включає:

- вхідний embedding-шар;
- двонаправлений LSTM-шар;
- два щільно зв'язані (Dense) шари з активацією ReLU;
- вихідний шар із сигмоїдною функцією.

Навчання проводилося у 10 епохах при batch size = 64. Для оптимізації використано Adam як оптимізатор та binary crossentropy як функцію втрат. Навчання супроводжувалося early stopping для уникнення перенавчання. Символи були векторизовані методом one-hot encoding, дані розділено у співвідношенні 70:15:15 на навчальну, валідаційну та тестову вибірки.

В результаті також було отримано метрики оцінки якості роботи моделі машинного навчання: Binary Cross-Entropy (далі BCE), accuracy, precision, recall та F1-score.

Метрика BCE – це функція втрат, тобто, що більша помилка, то більша "кара". Розраховується дана метрика за формулою

$$BCE = -\frac{1}{N}\sum_{i=1}^{N}[y_i * log(\hat{y}_i) + (1 - y_i) * log(1 - (\hat{y}_i)] \quad (4)$$

де y_i — справжня мітка: 1 (є частина імені/слова) або 0 (не є), $\hat{y}_i$ — ймовірність, передбачена моделлю (рис. 1).

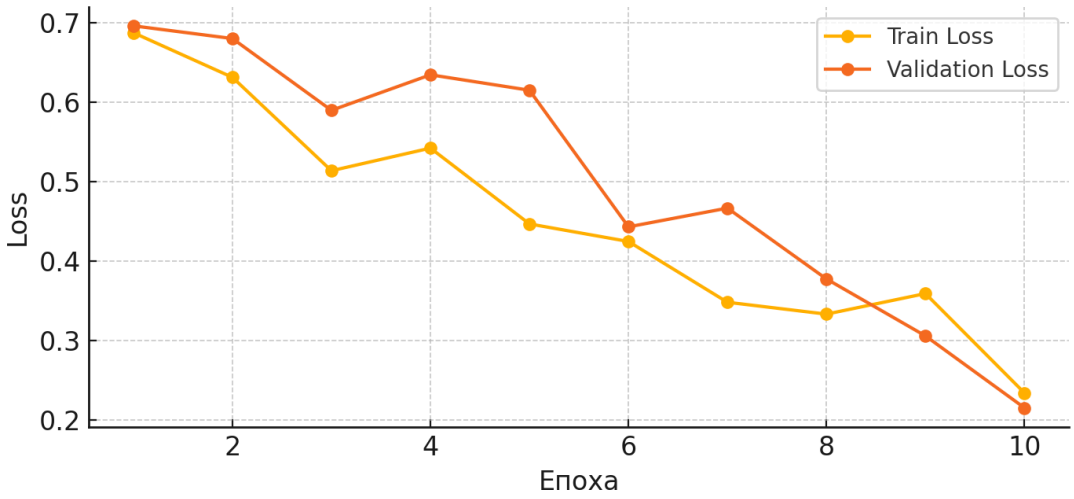


Рис. 1. Графік втрат під час навчання моделі

Метрика Accuracy відповідає за точність моделі і показує наскільки в цілому добре модель відгадує, розрахунок проводиться за наступною формулою

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

де TP (true positive) — правильно вгадано, що це частина імені/слова, TN (true negative) — правильно вгадано, що не частина імені/слова, FP (false positive) — помилково сказано, що це ім'я/слово, FN (false negative) — не розпізнано частину імені/слова. Графік точності моделі наведено на рисунку 2.

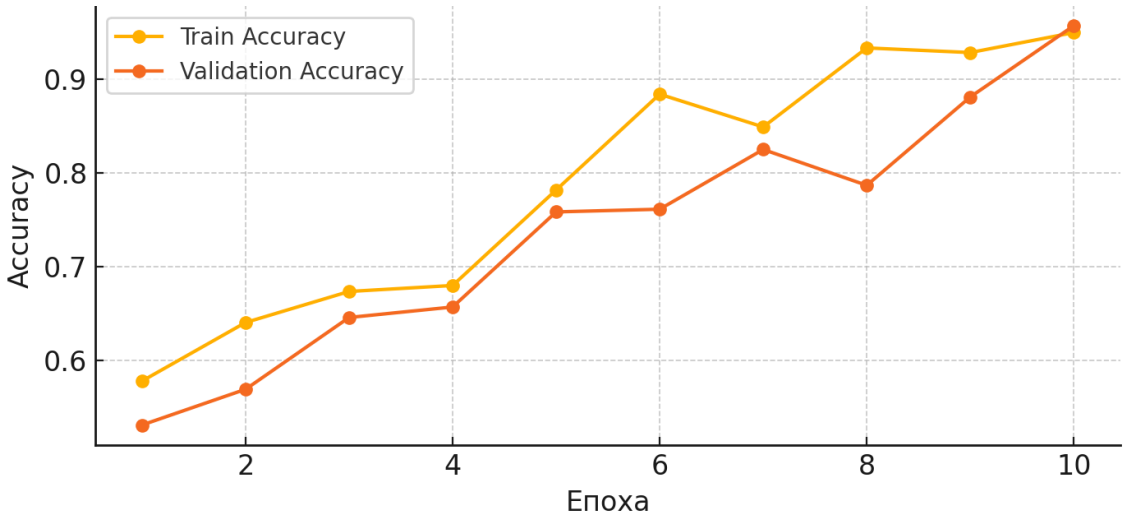


Рис. 2. Графік точності моделі

Precision показує частку правильних позитивних передбачень з усіх передбачень, які модель вважає позитивними. Тобто, скільки з тих символів, які модель вважає "іменами", насправді є іменами. Дана метрика розраховується за виразом, наведеним нижче

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

де TP (true positive) — правильно вгадано, що це частина імені/слова, FP (false positive) — помилково сказано, що це ім’я/слово. Графік влучності для запропонованої моделі показано на рисунку 3.

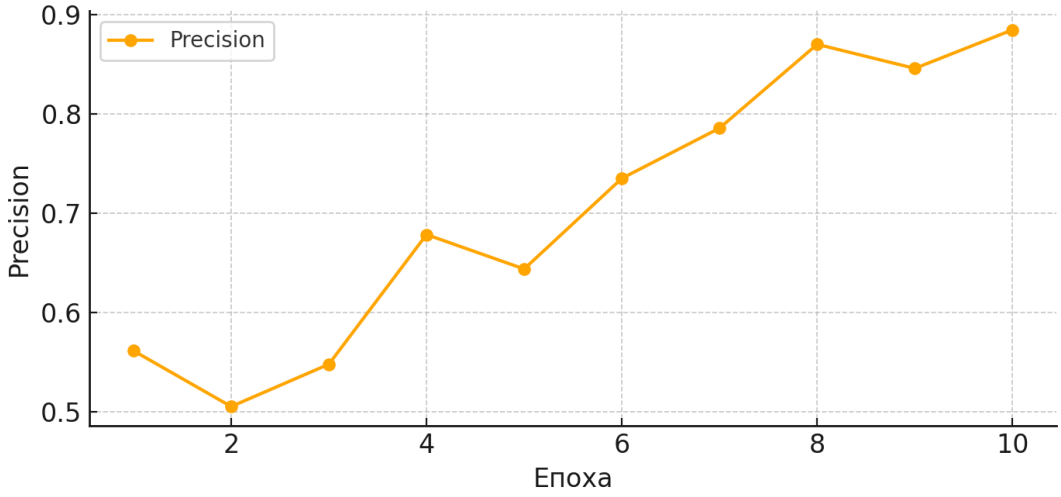


Рис. 3. Графік влучності моделі

Recall – це метрика, яка показує частку всіх справжніх позитивних прикладів, які модель виявила. Тобто, з усіх символів, які справді були іменами, скільки модель змогла знайти. Цей показник обчислюється за наступним виразом

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

де TP (true positive) — правильно вгадано, що це частина імені/слова, FN (false negative) — не розпізнано частину імені/слова. Графік метрики recall для запропонованої моделі наведено на рисунку 4.

Рис. 4. Графік чутливості моделі

Метрика F1-score є гармонійним середнім між precision і recall. Вона дає уявлення про те, наскільки добре модель справляється з виявленням позитивного класу, враховуючи і точність (precision), і повноту (recall), особливо коли дані є незбалансованими. Дана метрика розраховується за наступною формулою

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{8}$$

Графік F1-score для запропонованої моделі наведено на рисунку 5.

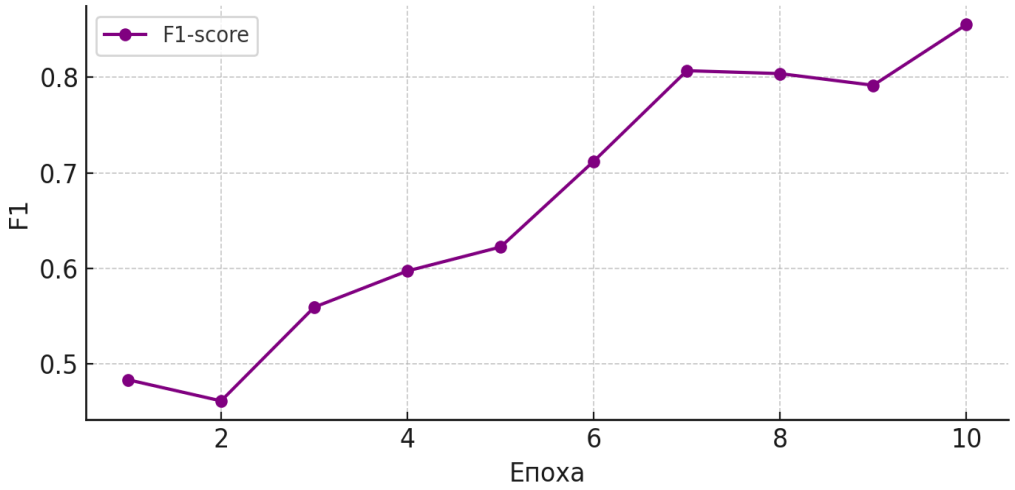


Рис. 5. Графік гармонійного середнього моделі

Після навчання модель інтегровано в додаток де вона аналізує наявність слів та імен в паролях, до основного пошуку дат, телефонних номерів, повторів, шаблонів клавіатурного введення, років. Оцінка видається у вигляді списку та пояснення: що саме робить пароль слабким. У разі потреби користувачеві пропонується варіант автоматично покращеного пароля.

Результати. Після завершення розробки програмного засобу KrystalLock було проведено серію тестувань, метою яких була перевірка відповідності роботи запропонованих методів початковим вимогам, а також оцінка їх ефективності. Аналіз проводився як у функціональному, так і в експериментальному аспектах, з метою виявлення практичної користі використаних алгоритмів.

Одним із ключових результатів стало підтвердження того, що реалізовані методи генерації забезпечують створення паролів із високими характеристиками ентропії та структурної різноманітності.

Зокрема, при використанні випадкової генерації (модуль secrets) із довжиною 16 символів і включенням чотирьох типів символів (великі, малі літери, цифри, спецсимволи), теоретична ентропія пароля перевищує 100 біт, що відповідає найвищому рівню захисту за сучасними стандартами. Аналіз сформованих паролів через зовнішні сервіси (VpnMentor, PeriTools) підтвердив, що 100% згенерованих комбінацій отримують найвищу оцінку безпеки.

Щодо системи аналізу стійкості, то основна увага була зосереджена на перевірці точності та коректності роботи нейромережевої моделі BiLSTM. Тестова вибірка містила понад 500 000 паролів. Модель досягла точності 92.3% з F1-метрикою 0.91, що свідчить про високу ефективність класифікації навіть за наявності неоднозначних шаблонів.

Як приклад:

– пароль London2025! було класифіковано як слабкий — модель виявила наявність назви міста та поточного року, що часто трапляється в злитих базах.

– пароль R#9cL^v7f@Px — класифіковано як сильний, оскільки не містить змістовних патернів, має високу ентропію та використовує різні категорії символів.

Окрему увагу було приділено поясненнями щодо слабких місць. Наприклад, при введенні Anna1998 система зазначає: «Ймовірне ім’я; дата народження; надто короткий; був знайдений у відкритих витоках».

Було також проведено порівняння з аналогами, включаючи вбудовані системи оцінки у Google, Microsoft, менеджери паролів (Bitwarden, 1Password). Згідно з результатами, KrystalLock більш точно класифікує семантично вразливі паролі, які традиційні rule-based системи вважають «середніми» або навіть «надійними» так як в них відсутні розроблені методи.

Висновок. У межах проведеної роботи було розроблено комплекс методів генерації та багаторівневого аналізу стійкості паролів, які ґрунтуються на поєднанні структурних, лексичних та евристичних ознак. Запропонований підхід інтегрує сучасні принципи кібербезпеки, когнітивні механізми формування мнемонічних паролів, а також інструменти глибинного машинного навчання для виявлення латентних вразливостей на семантичному рівні.

Ключовою особливістю розроблених методів є комплексність та гнучкість: система оцінки стійкості паролів поєднує класичні критерії (довжина, символьний склад, відповідність словникам скомпрометованих комбінацій) з аналізом шаблонів, виявленням поширених лінгвістичних структур і застосуванням нейромережевої моделі на основі архітектури BiLSTM. Це дозволяє ефективно класифікувати паролі за рівнем безпеки навіть у випадках, коли традиційні rule-based алгоритми не виявляють ризиків. Модель продемонструвала високу точність класифікації (92.3%) за метриками F1, Precision та Recall, що підтверджує ефективність використаного підходу.

У рамках генерації паролів розроблено три самостійні, але взаємопов’язані методи: випадковий, мнемонічний та комбінований. Випадковий метод базується на криптографічно стійкому виборі символів з гарантованим покриттям усіх обраних категорій. Мнемонічний метод формує паролі на основі перетворення змістовних фраз за чітко визначеними правилами з використанням вибіркових leet-замін, що забезпечує баланс між запам’ятовуваністю і складністю. Комбінований підхід синтезує переваги обох зазначених методів, додаючи до мнемонічної основи випадкові елементи для підвищення ентропії та руйнування прогнозованих шаблонів.

Таким чином, результати підтверджують, що розроблені методи забезпечують як надійність створюваних паролів, так і глибину їхнього аналізу відповідно до сучасних викликів кібербезпеки. Запропоновані рішення можуть бути використані як основа для створення більш гнучких, адаптивних та інтелектуальних систем управління

автентифікаційними даними, а також сприяти формуванню нових підходів у сфері персоналізованого захисту цифрових ідентичностей.

Список літератури

1. The Tangled Web of Password Reuse. https://arxiv.org/pdf/1706.01939
2. NIST SP 800-63B:2017. Керівництво з цифрової ідентифікації. Аутентифікація та управління життєвим циклом. [Чинний від 2017]. Вид. офіц. Гейтесбург (США), 2017. 74 с.
3. Buchanan W. J. Applied Cryptography and Network Security: Modern Implementations in Python. London: Springer Nature. 2022. 428 с.
4. Chapple M., Seidl D. *CISSP (ISC)* Certified Information Systems Security Professional. Hoboken (USA): Wiley. 2018. 191с.
5. Заяц І. М. Кібергігієна: навчальний посібник. Київ: НА СБУ. 2021. 66 с.
6. Samarati P., De Capitani di Vimercati S. Security and Privacy in Password-Based Authentication: Risks and Countermeasures. New York (USA): ACM Computing Surveys. 2017. 49 с.
7. Bertino E., Sandhu R. Handbook of Information Security Management. Boca Raton (USA): CRC Press, 2018. 38 с.
8. Password Strength Checker Test Your Password Security. https://www.idstrong.com/tools/password-strength-checker/
9. Molich R., Nielsen J. Improving a Human-Computer Dialogue. New York (USA): Communications of the ACM, 2019. 80 с.
10. Balancing Password Security and User Convenience: Exploring the Potential of Prompt Models for Password Generation. https://www.mdpi.com/2079-9292/12/10/2159
11. Password Strength Checker. https://peritools.com/password-strength-checker
12. Ney P., Paz C., Alharthi A. Password Strength Analysis Using Machine Learning Approaches: An Open-source Tool. International Journal of Cyber Security and Digital Forensics, 2019, Vol. 8(4). № 4. С. 222–235.
13. Sweigart A. Beyond the Basic Stuff with Python: Best Practices for Writing Clean Code. San Francisco: No Starch Press, 2020. 384 с.
14. A Deep Learning Approach for Password Guessing. / Hitaj B., Gasti P., Ateniese G., Perez-Cruz F. PassGAN. URL: https://arxiv.org/abs/1709.00440
15. Прикладна математика та комп’ютинг ПМК'2022: Збірник тез доповідей. Київ. 2022. С. 6. URL: https://pzks.fpm.kpi.ua/documents/pmk/PMK2022.pdf

Т.Ю. Кришталь, О.В. Троянський, Н.І. Кушніренко

METHODS OF GENERATION AND ANALYSIS OF PASSWORD STRENGTH TAKING INTO ACCOUNT LEXICAL, STRUCTURAL AND HEURISTIC CHARACTERISTICS

T.Yu. Kryshtal, O.V. Troyanskiy, N.I. Kushnirenko

National Odesa Polytechnic University
1, Shevchenko Ave., Odesa, 65044, Ukraine
Emails: kushnirenko@op.edu.ua, o.v.troyanskiy@op.edu.ua

The article discusses the developed methods for generating and analyzing password strength, taking into account lexical, structural, and heuristic features, which were implemented within the framework of the development of the KrystalLock software product. The main goal of the work was to create effective methods that ensure both generation and verification of passwords for compliance with modern security requirements. Within the framework of the implemented approach, three methods of password generation are proposed: random, mnemonic, and combined. Mnemonic generation is based on creating passwords from phrases that are easy for the user to remember, with the subsequent use of selective leet transformations to increase complexity. All methods support parameter settings, in particular, password length, the use of separate categories of characters (upper/lower case, numbers, special characters), as well as control over meaningful templates. In the part of password strength analysis, a multi-level method has been implemented that combines classical complexity assessment criteria with modern approaches to machine analysis. In particular, the length, character composition, pattern detection (repetition, sequences, common keyboard combinations), checking for the presence of known merged password databases, as well as lexical and semantic analysis are carried out. Special attention is paid to the machine approach to weak password detection, which is built on the bidirectional long-term memory model (BiLSTM). The model is trained on a dataset of passwords containing various names and words and demonstrates accuracy of over 92% accuracy on test samples, using the F1, Precision and Recall metrics. The developed application is implemented in the Python environment using the TensorFlow, NumPy, secrets and other libraries. Comprehensive testing of the developed methods was carried out, including both functional and load scenarios. The compliance of the analysis results with the expected cybersecurity standards (NIST, OWASP, ISO/IEC 27001) was assessed and the effectiveness of the approach was confirmed compared to existing analogues. The results of the work can be used to increase the level of account protection in personal and corporate environments, as well as for implementation in password management systems. The application can become the basis for further research and improvement in the field of adaptive authentication, in particular by integrating behavioral analysis or biometric factors. The complexity of the approach, the use of deep learning models and the possibility of personalization make the developed product a valuable contribution to the field of practical cybersecurity.
Keywords: password strength, password generation, machine learning, BiLSTM, mnemonic generation, semantic analysis, cybersecurity.

DOI 10.15276/imms.v15.no2.229
УДК 004.056.53

Informatics and Mathematical Methods in Simulation
Vol.15 (2025), No. 2, pp. 229-239

РОЗРОБКА ЗАСТОСУНКУ ДЛЯ ПОШУКУ ПРОФІЛІВ КОРИСТУВАЧІВ У СОЦІАЛЬНИХ МЕДІА ЯК ПОЧАТКОВИЙ ЕТАП OSINT-ДОСЛІДЖЕННЯ

А.В. Котляров, Н.І. Кушніренко, В.О. Назаров

Національний університет «Одеська політехніка»
1, Шевченка пр., м.Одеса, 65044, Україна
Email: kushnirenko@op.edu.ua

У статті розглянуто розробку застосунку для пошуку профілів користувачів у соціальних медіа в контексті OSINT-дослідження. Досліджено основні типи ідентифікаторів користувачів, за якими можна здійснювати такий пошук, зокрема: номер телефону, адреса електронної пошти, зображення, прізвище та ім'я, нікнейм. Детально проаналізовано переваги та обмеження кожного з цих ідентифікаторів. Розроблено та описано алгоритми автоматизованого пошуку за прізвищем та ім'ям, а також за нікнеймом з відповідними блок-схемами. Описано основні технічні проблеми, що виникають при автоматизації процесу, включаючи обробку CAPTCHA, і запропоновано шляхи їх вирішення за допомогою бібліотеки Selenium WebDriver, що дозволяє емулювати дії реального користувача. На основі розроблених алгоритмів створено програмний застосунок "KotOSINT" мовою Python з графічним інтерфейсом на основі Tkinter. Застосунок забезпечує функціонал пошуку профілів користувачів у різних соціальних медіа, можливість додавання нових платформ, збереження історії пошуків та їх експорт. Проведено тестування, яке показало середній час перевірки одного посилання близько 4,2 секунди, що свідчить про ефективність застосунку.
Надано практичні рекомендації щодо безпечного використання розробленого інструменту, зокрема щодо застосування VPN, створення окремих профілів браузера з фейковими акаунтами, врахування мовних налаштувань системи та попереднього тестування в режимі перевірки. Застосунок "KotOSINT" має потенціал стати важливим інструментом у протидії кіберзлочинності та підвищенні ефективності OSINT-дослідження у сучасному інформаційному середовищі, де соціальні медіа відіграють ключову роль у комунікації та поширенні інформації. Розробка подібних інструментів є особливо актуальною в контексті збільшення кіберзагроз та необхідності розвитку засобів цифрової розвідки.
Ключові слова: OSINT, соціальні медіа, браузер, профіль користувача, нікнейм, Selenium.

Вступ. У сучасному цифровому світі соціальні медіа стали невід'ємною частиною повсякденного життя мільярдів людей, забезпечуючи швидку комунікацію та соціальну взаємодію. За даними Statista, станом на 2024 рік понад 5 мільярдів людей у світі користуються ними, що становить майже 62% глобального населення, і очікується, що до 2028 року ця кількість зросте до понад 6 мільярдів користувачів. При цьому середня тривалість щоденного користування становить 151 хвилину [1].

Соціальні медіа – це вид масмедіа, побудованих на ідеологічних та технологічних принципах Web 2.0. Перші соціальні медіа почали з'являтися ще у 1990-х роках, проте основний етап їх розвитку розпочався у 2004 році, коли Тім О'Рейлі опублікував статтю "What Is Web 2.0", що окреслила ключові принципи нової цифрової епохи [2]. Зараз соціальні медіа охоплюють широкий спектр ресурсів, які умовно можна поділити за їхнім призначенням на такі типи: соціальні мережі, медіа-платформи, блоги, форуми, освітні ресурси, месенджери, платформи для знайомств, вікі-проєкти та віртуальні ігрові світи [3].

Проте соціальні медіа, у поєднанні з анонімністю, стали основним середовищем для здійснення кіберзлочинів. Серед прикладів такого використання – пропаганда, тероризм, наркоторгівля, шахрайство, порнографія, піратство, кібербулінг. Водночас

особливою загрозою сьогодення є їхнє використання у військовому контексті, що може призвести до витоку критично важливої інформації та поставити під загрозу національну безпеку [4].

У боротьбі з цими загрозами важливу роль відіграє OSINT (Open Source Intelligence), який є одним із трьох ключових напрямів сучасної розвідки. Під OSINT розуміють збір, оцінку та аналіз інформації із загальнодоступних джерел (як платних, так і безкоштовних) з метою їх використання як розвідданих або доказів [5]. Засновником даної технології вважається Шерман Кент, який тривалий час працював в Центральному розвідувальному управлінні (ЦРУ). Саме під його керівництвом був створений «Єльський звіт», який спричинив неабиякий резонанс у наукових та військових колах. Документ оцінював, яку інформацію про боєготовність американських збройних сил противник міг зібрати з відкритих джерел. В результаті понад 90% зібраної інформації виявилась правдивою [6].

Застосування OSINT має низку важливих переваг порівняно з іншими видами розвідки, зокрема: простоту отримання інформації, швидкість збору та аналізу даних, безпеку розслідувань, економічну ефективність та можливість моніторингу в реальному часі. Що робить його надзвичайно цінним для правоохоронних органів, журналістів, військових та інших спеціалістів із кібербезпеки. Зважаючи на це, виникає потреба у розробці спеціалізованих застосунків, які дозволятимуть здійснювати пошук потенційних зловмисників за їхніми цифровими слідами.

Мета і задачі дослідження. Метою роботи є розробка застосунку для пошуку профілів користувачів у соціальних медіа, що дозволить ефективно виявляти можливі профілі реальних злочинців. Для досягнення поставленої мети були визначені наступні задачі:

- дослідити основні типи ідентифікаторів користувачів;
- розробити алгоритм пошуку профілів користувачів у соцмедіа;
- реалізувати застосунок, на основі даного алгоритму,
- провести тестування застосунку та надати рекомендації щодо використання.

Основна частина. Для пошуку інформації в соціальних медіа потрібна відправна точка – початкові дані, за якими здійснюватиметься пошук. Найважливішими серед них є ідентифікатори – унікальні або частково унікальні відомості, що дозволяють точно визначити особу або звузити коло пошуку. Основними відомостями, за якими можна знайти профілі в соціальних медіа, є:

- номер телефону;
- адреса електронної пошти;
- зображення;
- прізвище та ім’я;
- нікнейм.

Перші два відносяться до категорії чітких ідентифікаторів, оскільки дозволяють однозначно встановити зв’язок між різними обліковими записами користувача. Це зумовлено тим, що вони є унікальними для кожного користувача, адже використовуються в процесі реєстрації або авторизації в соцмедіа і не можуть повторюватися іншими користувачами. Однак у більшості випадків ці ідентифікатори не відображаються відкрито на самій платформі, що обмежує їх використання для пошуку. Проте навіть якщо це можливо, користувач може обмежити видимість цих даних у налаштуваннях конфіденційності. Прикладами є месенджери Telegram і Viber, в яких можна знайти профіль користувача, якщо додати його номер до списку контактів. Нажаль повноцінна автоматизація цього процесу ускладнюється через високу ймовірність блокування.

У зв’язку з обмеженими технічними можливостями відкритого пошуку за цими ідентифікаторами, на практиці вони найчастіше застосовуються для пошуку в DarkNet, зокрема по злитих базах даних. Хоча такі джерела також можуть використовуватись в

OSINT, у межах даної статті подібні методи не реалізовуватимуться з огляду на їхню сумнівну законність та невідповідність етичним принципам.

Наступним розглянемо – зображення. Воно не є настільки чітким ідентифікатором, як попередні, особливо у випадках, коли зображення не є особистим. Навіть якщо на фотографії присутнє обличчя, це не завжди дозволяє достовірно встановити зв'язок із певним профілем, оскільки користувач може завантажити будь-яке зображення у власний обліковий запис.

Ще одна проблема полягає в тому, що соцмедіа не мають вбудованої можливості пошуку за зображеннями. Натомість існують сторонні сервіси, що мають величезні власні бази даних із застосуванням складних алгоритмів розпізнавання облич. Через те, що ці інструменти працюють незалежно від самих соцмедіа і спираються на власні технології, реалізація пошуку за таким типом ідентифікатора не розглядатиметься в даній роботі. Проте варто знати, що такі сервіси, як Google Images, PimEyes і Betaface, є корисними у процесі OSINT-дослідження.

Розглянемо пошук за прізвищем та ім'ям (далі ПІ). ПІ є однією з найочевидніших та найпростіших точок входу для пошуку профілів. Це пояснюється тим, що під час реєстрації всі платформи вимагають вказувати їх, і вони завжди залишаються відкритими для перегляду, навіть якщо профіль повністю закритий.

Однак слід зазначити, що такий пошук не є чітким, оскільки в соціальних медіа може існувати безліч людей із таким самим прізвищем та ім'ям. Зрозуміло, що чим більш рідкісне ПІ тим менша кількість таких профілів, що полегшує подальшу ідентифікацію особи. Ще однією проблемою є різні варіації написання, пов'язані з скороченням, транслітерацією, перекладом та заміною символів.

Тим не менш, використання цього ідентифікатора дозволяє визначити хоча б те, чи є на певній платформі користувач із такими даними.

Щоб здійснити пошук профілю «вручну», достатньо скористатися вбудованим пошуком в будь-якій соцмедіа. Для цього потрібно лише ввести ім'я та прізвище у відповідне пошукове поле, а потім переглянути результати. Якщо потрібно, скористатися фільтрами.

Автоматизувати пошук за цим ідентифікатором можна через використання пошукових URL в самих соціальних медіа. Наприклад, для LinkedIn використовується формат – «https://www.linkedin.com/search/results/people/?keywords={}». Куди замість символів «{}» підставляється будь-яке ПІ. Для визначення наявності профілю буде здійснюватися пошук певної фрази-помилки на сторінці. У випадку LinkedIn це фраза «No results found». Якщо цей текст присутній у відповіді сервіса, це означає, що профілів із такими даними не знайдено. Але також слід враховувати мову, на якій будуть відображатися дані фрази. На рисунку 1 зображено блок-схему алгоритму пошуку за ПІ у розроблюваному застосунку.

Ще одним видом пошуку є пошук за Нікнеймом. Нікнейм (від англ. nickname – прізвисько) – це вигадане ім'я, яке користувач використовує в соціальних медіа. Зазвичай воно походить від власного імені чи прізвища, може бути натхненне вигаданими персонажами, відомими особистостями або мати інше значення, яке пов'язане з власником. У соцмедіа нікнейм виконує функцію унікального ідентифікатора користувача, замінюючи числовий ID. Це забезпечує зручність у спілкуванні та взаємодії, адже запам'ятати нікнейм набагато простіше, ніж довгий набір цифр. Завдяки тому, що на одній платформі кожен нікнейм є унікальним, він забезпечує більш точну ідентифікацію профілю порівняно з ПІ. Водночас існує ймовірність, що той самий нікнейм буде використаний іншою особою на іншій платформі. Також є проблема з можливістю його зміни на інший, що може викликати додаткові труднощі.

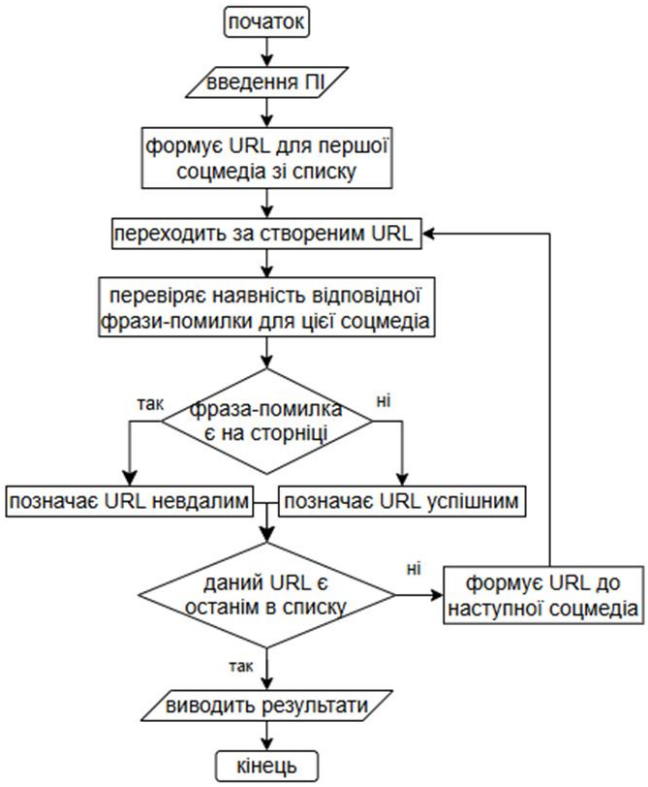


Рис. 1. Блок-схема алгоритму пошуку за прізвищем та ім’ям

Для багатьох користувачів характерна лінь, що проявляється у небажанні створювати окремі нікнейми для кожної соціальної медіа – аналогічна поведінка спостерігається і щодо паролів. Цей фактор, поряд із прагненням до зручності, призводить до повторного використання одного і того самого нікнейму на різних платформах.

Частою також є практика використання частини перед «@» в адресі електронної пошти. Наприклад, якщо суб’єкт використовує «mpulido007@gmail.com», то існує велика ймовірність, що він може використовувати «mpulido007» як псевдонім в якійсь із соціальних медіа [7].

Здійснення пошуку «вручну» за нікнеймом є подібним до пошуку за прізвищем та ім’ям: користувач вводить запит у пошукове поле в соцмедіа та отримує результат. Проте слід враховувати, що не всі підтримують пошук саме за нікнеймом.

Автоматизувати пошук можна два способами. Перший – простіший у реалізації, відповідно більш швидший. Зазвичай використовується лише бібліотека Requests, яка дозволяє отримувати HTTP-статус сторінки. Якщо код 2xx, то профіль існує, якщо 4xx то ні. Однак у цього метода проблеми з надійністю перевірки, оскільки не всі соцмедіа повертають 4xx навіть у разі відсутності користувача, а замість цього вони можуть просто перенаправляти запит на головну сторінку або ж повертати код 200 з відображенням текстової помилки (так звана м'яка помилка 404 – soft 404). Через це алгоритм може помилково вважати, що профіль існує, коли насправді його немає.

Другий спосіб повільніший, однак майже стовідсотково визначить чи є профіль в конкретній соцмедіа чи ні. Цей спосіб базується на поєднанні бібліотек Requests та BeautifulSoup. Спочатку Requests визначає HTTP-статус, при цьому додатково проводиться перевірка на 3xx коди, тобто на перенаправлення на іншу сторінку. Після цього, якщо отримано код 2xx, BeautifulSoup починає шукати відповідну фразу-помилку на сторінці.

Прикладом реального застосунку, який використовує подібний алгоритм пошуку є Sherlock [7]. Розроблюваний застосунок також буде використовувати вище наведений алгоритм. На рисунку 2 зображена схема алгоритму за нікнеймом.

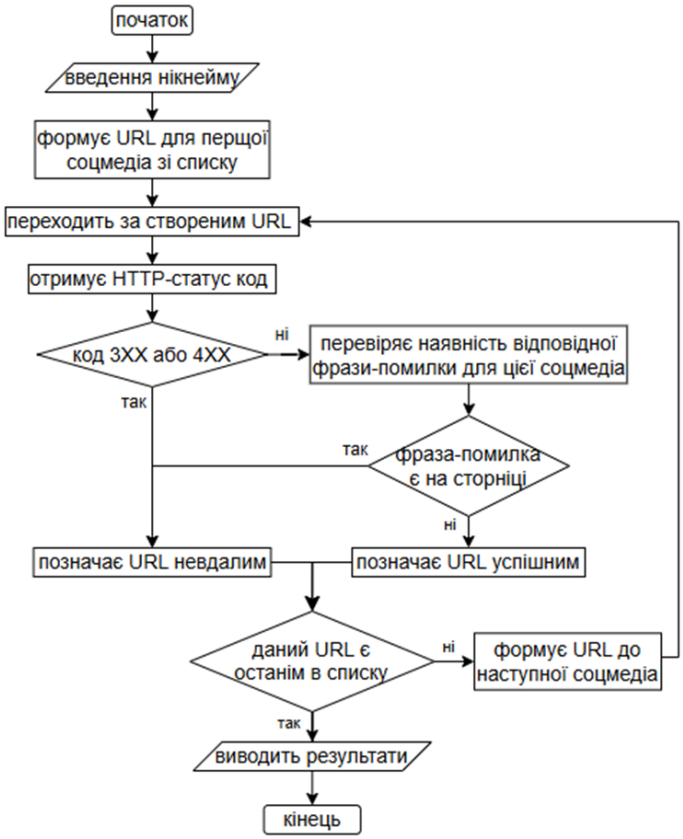


Рис. 2. Блок-схема алгоритму пошуку за нікнеймом

Отже, розроблюваний застосунок реалізовуватиме механізми пошуку за двома типами ідентифікаторів користувача – прізвищем і ім'ям, а також за нікнеймом.

Водночас описані вище алгоритми для обох типів, мають спільне суттєве обмеження – вони не передбачають обробку CAPTCHA, що є важливою перешкодою для автоматизації процесу. CAPTCHA – це технологія, призначена для відокремлення людських користувачів від автоматизованих ботів. Найпоширенішим методом їх автоматизованого вирішення є використання онлайн-сервісів розв'язання CAPTCHA. Інший метод передбачає використання ротації IP-адрес. Однак дані методи не є безкоштовними [8].

Існує ще один метод, який дозволить зменшити ймовірність появи CAPTCHA майже до нуля. Якщо ж вона навіть з'явиться, то перевірка на існування профілю відбудеться швидше за її появу. Реалізується це за допомогою бібліотеки Selenium WebDriver – це комплексний набір інструментів для автоматизації взаємодії з веб-браузерами, що дозволяє програмно керувати поведінкою браузера, імітуючи дії користувача на веб-сторінках. Тобто сайт сприйматиме запити від застосунку як звичайну активність реального користувача, а не як автоматизований скрипт.

Водночас можливе використання будь-якого браузерного профілю, а не лише гостьового режиму. Це відкриває додаткові можливості для обходу обмежень, пов'язаних із необхідністю обов'язкової авторизації у деяких соціальних медіа, без якої доступ до певного контенту є неможливим. Завдяки використанню збережених облікових даних у профілі браузера, автоматизована система може діяти від імені реального користувача, що дозволяє уникнути появи сторінок авторизації.

Тому саме цей метод обрано для розроблюваного застосунку.

Однак даний підхід не позбавлений певних недоліків. По-перше, використання Selenium WebDriver створює додаткове навантаження на обчислювальні ресурси системи. По-друге, загальний час виконання перевірок суттєво зростає. І останній,

найбільш суттєвий недолік – це потенційна загроза конфіденційності. Оскільки Selenium WebDriver має доступ до всього функціоналу браузера, автоматизована програма може, прямо чи опосередковано, отримати доступ до чутливої інформації. У випадку компрометації такого коду, це може створити ризик витоку даних користувача. Водночас, це питання легко вирішується – достатньо попередньо створити окремий браузерний профіль без будь-яких чутливих даних із фальшивими обліковими записами для входу у необхідні соцмедіа. Таким чином, користувач може підготувати середовище для запуску заздалегідь і не перейматися за свою конфіденційність, повністю контролюючи, які саме дані використовуються під час роботи застосунку. Додатково можна використовувати VPN. Проте, для досягнення максимального рівня безпеки, рекомендується поєднувати зазначені заходи з ізоляцією середовища запуску.

Розроблюваний застосунок має назву «KotOSINT». За замовчуванням у застосунку передбачено наявність 50 URL для пошуку за нікнеймами та 25 – для пошуку за ПІ. Кожен з них супроводжується відповідними фразами-помилками англійською та українською мовами.

У якості основної мови програмування обрано Python, через його зрозумілий синтаксис та розвинену екосистему бібліотек. Середовищем розробки є PyCharm. Для побудови графічного інтерфейсу використано бібліотеку Tkinter.

Для збереження даних прийнято рішення використовувати JSON-файли замість традиційної бази даних. Це зумовлено низкою факторів, серед яких невеликий обсяг даних, їх простота та відсутність необхідності в складних структурах або взаємозв'язках. Тому використання традиційної бази даних було б надмірним для такого застосунку. Однак при потребі структура JSON дозволяє без зусиль перенести дані до низки сучасних СУБД, зокрема таких як MongoDB, PostgreSQL або SQLite [9]. На рисунку 3 представлена іконка файлу запуску програми (batch-файл).

Рис. 3. Вигляд іконки файлу запуску застосунку

Після запуску batch-файлу відкриється вступне вікно (вкладка «Головна»), де представлено основні можливості застосунку. У цьому ж вікні зазначено контакт для зв'язку з розробником у разі виникнення питань. Дане вікно зображено на рисунку 4.

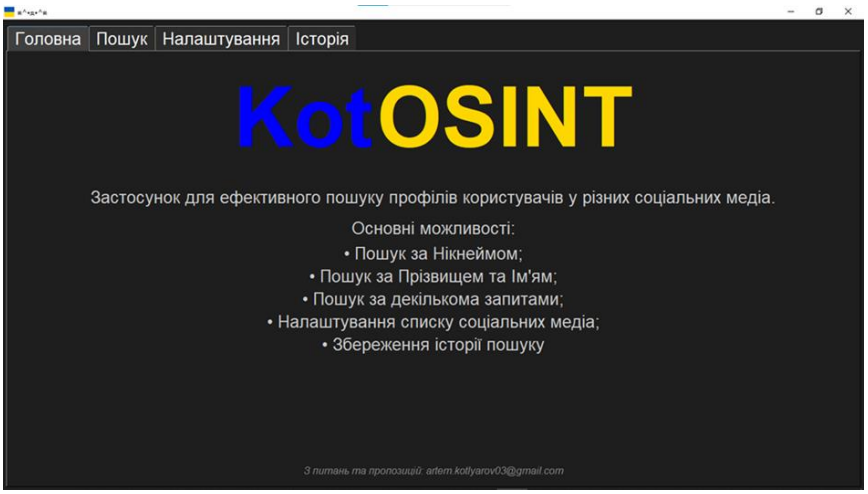


Рис. 4. Вступне вікно застосунку

Вкладка «Налаштування» має три розділи: «Профіль браузера», «URL для Нікнеймів» та «URL для ПІ».

У розділі «Профіль браузера» користувач має змогу вибрати бажаний браузер (Chrome, Edge, Firefox) і вказати назву профілю. Після цього достатньо натиснути кнопку «Зберегти». Обрані параметри будуть збережені, тож при наступних запусках програми їх не доведеться вводити знову. Якщо цього не зробити, то запуститься браузер Edge з Default профілем. На рисунку 5 зображено даний розділ.

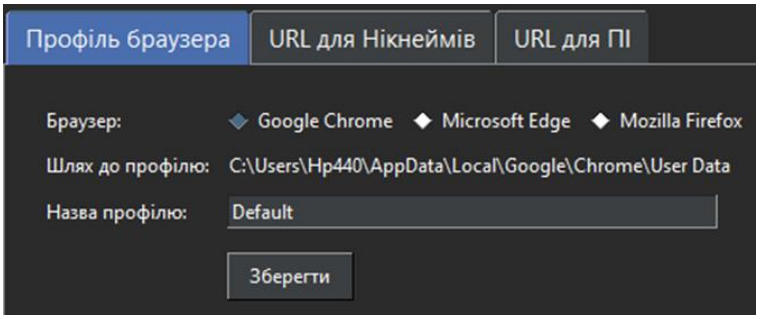


Рис. 5. Розділ обрання профілю браузера

Розділи «URL для Нікнеймів» та «URL для ПІ» загалом схожі. У верхній частині в обох розташована форма для додавання нових соціальних медіа. Де потрібно ввести: назву соцмедіа, URL (обов'язково із символами {}) та фрази-помилки (для ПІ це обов’язково). На рисунку 6 зображено ці форми з відповідними параметрами.

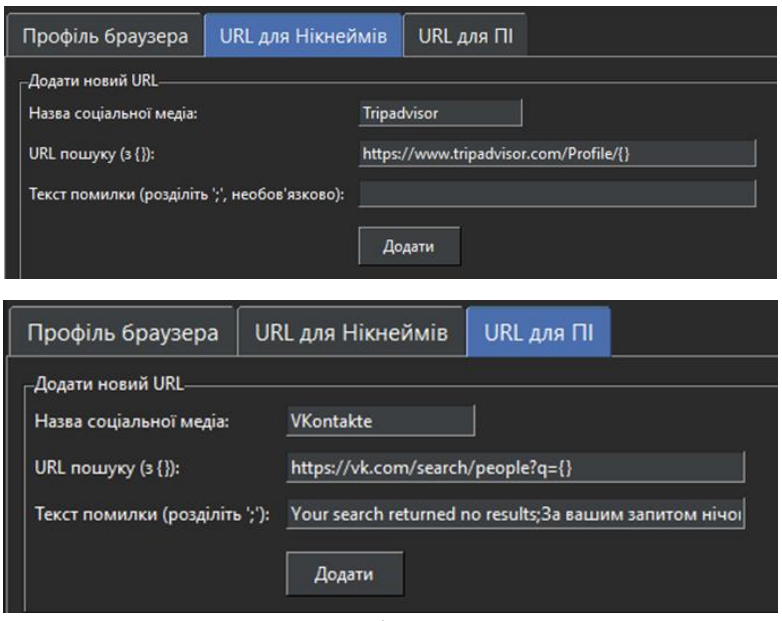


Рис. 6. Форми для додавання нових соцмедіа

Після натискання на кнопку «Додати» ці URL з’являться у списках, що знаходяться у нижній частині розділів. Якщо натиснути двічі на певний запис, то відкриється вікно редагування, де користувач може змінити будь-який параметр. Також, за необхідності, їх можна видаляти натисканням правої клавіші миші,.

Вкладка «Пошук» складається з двох частин. У правій частині, розміщено списки соціальних медіа, де можна вибрати потрібні варіанти, активуючи відповідні мітки. Для спрощення процесу вибору є кнопки «Вибрати всі» та «Зняти вибір» Крім того, реалізовано функціонал перетягування елементів у списку. Після вибору необхідних пунктів або всіх одразу слід натиснути кнопку «Зберегти». Якщо не вибрати жодного, то

під час спроби виконати пошук з’явиться повідомлення про помилку. На рисунку 7 зображено частину одного зі списків.

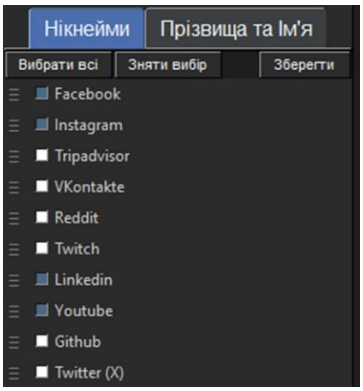


Рис. 7. Частина списку вибору соціальних медіа

У лівій частині даної вкладки знаходиться головні елементи інтерфейсу пошуку:

- кнопки вибору типу пошукового запиту («За Нікнеймом» або «За ПІ»);
- поле для введення цього запиту;
- кнопка для додавання додаткових полів введення (максимум 10), а також їх видалення;
- мітка «Режим перевірки», яка визначає чи виконуватиметься пошук у фоновому режимі, чи у інтерактивному режимі;
- кнопка «Пошук», яка розпочинає пошук;
- кнопка «Зупинити», яка зупиняє пошук в аварійному режимі;
- прогрес-бар, що відображає стан пошуку;
- поле для виведення отриманих результатів.

Для прикладу виберемо пошук за Нікнеймом та задамо такі значення: MaxTech01, MaxTech02, MaxTech03. Використаємо список соцмедіа, наведений на рисунку 7. Натискаємо на кнопку «Шукати». Після виконання пошуку отримуємо результат, що зображено на рисунку 8.

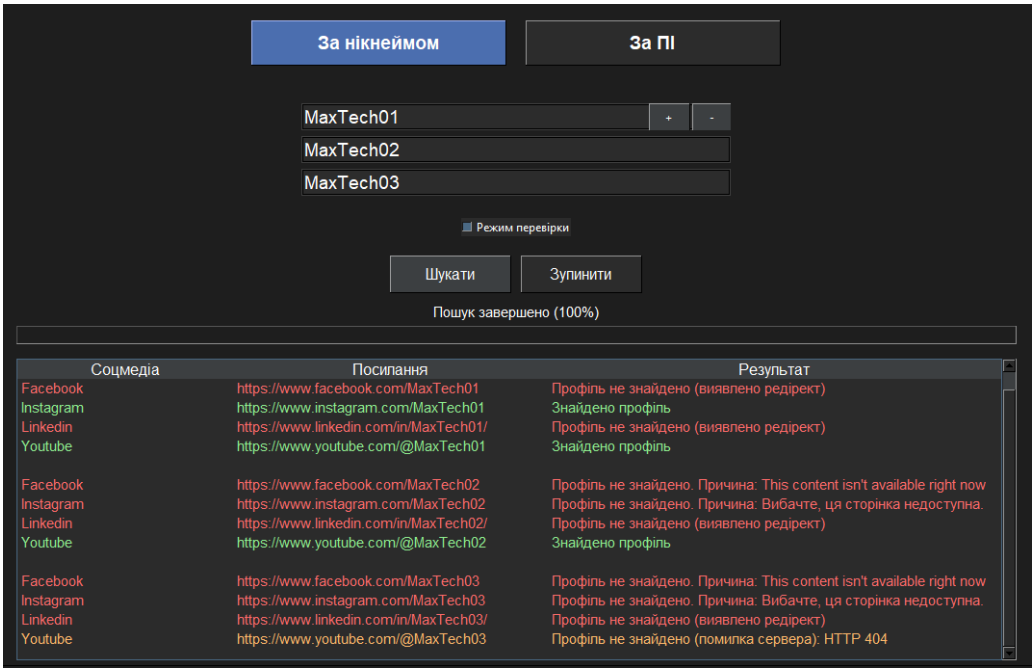


Рис. 8. Результати пошуку

Вкладка «Історія», відображає журнал попередніх пошукових запитів, що дозволяє користувачеві переглядати результати, отримані під час попередніх сеансів роботи із застосунком. Вона впорядкована за часом, тобто: найновіші результати розміщуються у верхній частині списку, тоді як більш давні – у нижній. На рисунку 9 зображено дану вкладку.

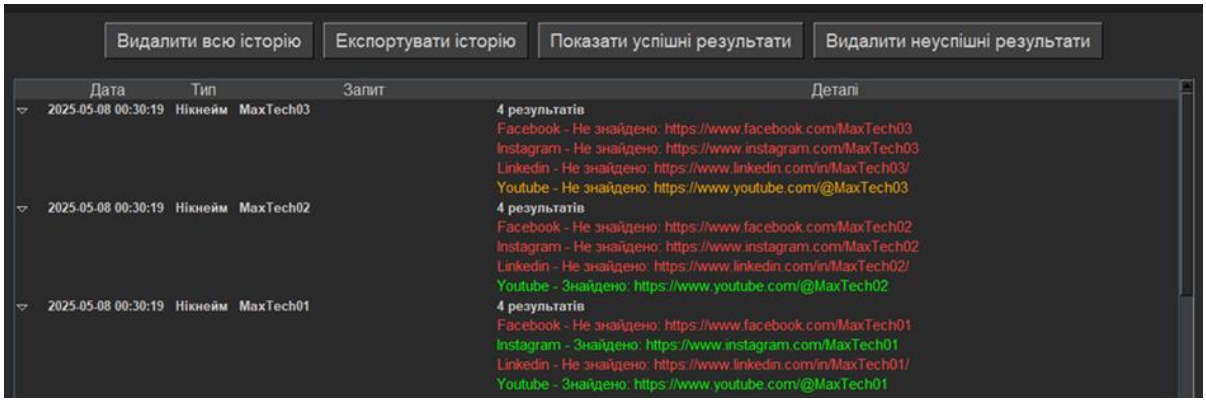


Рис. 9. Вкладка «Історія»

У верхній частині вкладки розміщені кнопки для виконання певних дій над історією, серед яких:

– «Видалити усі результати» – виконує повне очищення історії без можливості відновлення;

– «Показати успішні результати» – фільтрує записи, внаслідок чого на екрані відображаються лише успішні;

– «Видалити неуспішні результати» – виконує безпосереднє видалення всіх неуспішних записів з історії.

– «Експортувати історію» – генерує файл у форматі Excel та автоматично завантажує його на комп’ютер користувача.

Додатково, користувач має можливість видаляти окремі записи з історії – для цього достатньо натиснути правою кнопкою миші на обраному записі та вибрати пункт «Видалити запис».

Щодо тривалості пошуку, результати якого наведено на рисунку 8, варто зазначити, що весь процес зайняв 55 секунд і складався з двох основних етапів. Зокрема, 5 секунд було витрачено на запуск браузера, а решту часу – 50 секунд – було витрачено на перехід за посиланнями та їх аналіз. При цьому усього перевірено 12 посилань.

Для оцінки ефективності роботи системи доцільно розрахувати середній час перевірки одного посилання – t. Цей показник можна обчислити, поділивши час, витрачений на перевірку (без урахування часу запуску браузера) – T, на кількість перевірених посилань N. У даному випадку: $t = T / N = 50 / 12 \approx 4.2$ секунди. Що є досить гарним результатом.

Слід зазначити, що швидкість обробки запиту різниться залежно від ситуації. У випадку отримання негативного HTTP-статус коду, перевірка відбудеться майже миттєво. Навіть при успішному існує певна невизначеність щодо часу обробки. Якщо система знаходить фразу-помилку на початку сторінки, це відбувається швидше, ніж коли така фраза розташована в кінці. А якщо фраза-помилка взагалі відсутня, то процес займе ще більше часу, оскільки програмі необхідно проаналізувати весь вміст сторінки.

Окрім безпосередньої розробки застосунку, було також підготовано набір рекомендації для користувачів щодо його ефективного використання.

По-перше, необхідно використовувати VPN, що працює на рівні всієї системи, а не як розширення браузера. Це необхідно не лише для забезпечення безпеки, але й для отримання доступу до ресурсів, які були заблоковані відповідно до Указу Президента

України №133/2017 «Про застосування персональних спеціальних економічних та інших обмежувальних заходів (санкцій)» [10]. У сучасних умовах такі ресурси мають величезне значення для OSINT.

По-друге, створити окремий профіль у браузері, що дозволяє уникнути змішування особистої інформації з даними, що використовуються в дослідженнях. Також у межах цього профілю необхідно створити фейкові акаунти в різних соціальних медіа. Це особливо важливо, оскільки деякі платформи можуть виявити підозрілу активність і заблокувати особистий акаунт через сприйняття його як бота. Крім того, є певні ресурси, що можуть показувати іншим користувачам, хто переглядав їх акаунт, що створює ризик для анонімності.

По-третє, необхідно враховувати мову інтерфейсу. Застосунок за замовчуванням розпізнає повідомлення про помилки у соцмедіа лише українською та англійською мовами. Тому для коректного визначення наявності чи відсутності профілю бажано, щоб мовні налаштування системи відповідали одній із цих мов. Якщо ж користувач працює з іншим мовним інтерфейсом, є два варіанти: або вручну додати відповідні фрази до налаштувань застосунку, або змінити мову інтерфейсу кожної соцмедіа на англійську або українську. Важливо: обов'язково вимкнути функцію автоматичного перекладу сторінок у браузері.

По-четверте, перший запуск застосунку слід виконувати у режимі перевірки. Це дозволяє протестувати роботу системи та переконатися, що всі попередні налаштування здійснено коректно: створено відповідний профіль у браузері з фейковими акаунтами, обрана відповідна мова інтерфейсу, а також, за потреби, чи активовано VPN. Найважливіше – цей режим дає змогу впевнитися у правильності визначення наявності або відсутності профілю в соціальній медіа. Режим перевірки також доцільно використовувати при додаванні нових платформ до системи, щоб перевірити, як саме здійснюється пошук по ним.

Висновки. У рамках даної роботи було створено застосунок «KotOSINT», що призначений для автоматизованого пошуку профілів користувачів у соціальних медіа як початкового етапу OSINT-дослідження. У застосунку реалізовано два основні типи пошуку – за прізвищем та ім'ям (ПІ) і за нікнеймом, для яких наведено та описано відповідні алгоритми з детальними блок-схемами.

Застосунок реалізовано мовою програмування Python із використанням бібліотеки Tkinter для створення графічного інтерфейсу. Для збереження конфігураційних даних та результатів роботи обрано формат JSON, який забезпечує простоту структури та зручність подальшої інтеграції з повноцінними СУБД при потребі. Значною технічною перевагою застосунку є впровадження бібліотеки Selenium WebDriver, що забезпечує емуляцію дій реального користувача під час роботи в браузері. Це дозволяє ефективно обходити труднощі, пов'язані з появою CAPTCHA та необхідністю проходження авторизації.

У роботі також надано практичні рекомендації щодо безпечного та ефективного використання застосунку.

Тестування продуктивності розробленого застосунку засвідчило, що середній час перевірки одного посилання становить приблизно 4,2 секунди. Такий результат є достатньо високим для програмних засобів даного класу та підтверджує його практичну цінність. Отже, застосунок «KotOSINT» може істотно оптимізувати процес OSINT-досліджень і має потенціал стати дієвим інструментом у протидії кіберзлочинності.

Список літератури

1. Dixon S.J. Number of social media users worldwide from 2017 to 2028. URL: https://surl.lu/bsejrt
2. O'Reilly, T. What is Web 2.0: design patterns and business models for the next generation of software. O'Reilly Media, 2005. URL: https://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html

3. Aichner T., Jacob F. Measuring the degree of corporate social media use. International journal of market research. 2015. 57 p. DOI: https://doi.org/10.2501/IJMR-2015-018
4. Кучій О., Фролова О. Використання соціальних медіа як інструменту сучасної гібридної війни. *Acta de Historia & Politica: Saeculum XXI.* 2023. С. 93-104. DOI: https://doi.org/10.26693/ahpsxxi2023.si.093
5. Housego J. Guidance on open source investigation/research. London : National Police Chiefs' Council, 2020. 14 p.
6. Kent S. The Yale Report. 1951. 627 p. URL: https://www.cia.gov/resources/csi/static/The-Yale-Report.pdf
7. 27. Bazzell M. OSINT techniques: resources for uncovering online information: № 10. 2023. 550 p. URL: https://surl.li/rymyrr
8. Харченко В. О. Оцінювання вразливостей системи кіберзахисту на основі оптичного тесту Тюрінга CAPTCHA: дипл. … бак. : КПІ. Київ, 2021. 48 с.
9. Pranisha R. JSON vs SQL: What's the Difference? URL: https://olibr.com/blog/json-vs-sql-whats-the-difference/
10. Про застосування персональних спеціальних економічних та інших обмежувальних заходів (санкцій) : Рішення РНБО України від 15.04.2021 : станом на 23 квіт. 2021 р. URL: https://zakon.rada.gov.ua/laws/show/n0030525-21#Text

DEVELOPMENT OF AN APPLICATION FOR SEARCHING USER PROFILES IN SOCIAL MEDIA AS THE INITIAL STAGE OF OSINT RESEARCH

A.V. Kotliarov, N.I. Kushnirenko, V.O. Nazarov

National Odesa Polytechnic University
1, Shevchenko Ave., Odesa, 65044, Ukraine
Email: kushnirenko@op.edu.ua

The article discusses the development of an application for searching user profiles in social media in the context of OSINT research. The main types of user identifiers by which such a search can be performed are studied, in particular: phone number, email address, image, last name and first name, nickname. The advantages and limitations of each of these identifiers are analyzed in detail. Algorithms for automated search by last name and first name, as well as by nickname with corresponding flowcharts are developed and described. The main technical problems that arise when automating the process, including CAPTCHA processing, are described, and their solution is proposed using the Selenium WebDriver library, which allows emulating the actions of a real user. Based on the developed algorithms, a software application "KotOSINT" was created in Python with a graphical interface based on Tkinter. The application provides the functionality of searching user profiles in various social media, the ability to add new platforms, save search history and export them. Testing was conducted, which showed an average time of checking one link of about 4.2 seconds, which indicates the effectiveness of the application. Practical recommendations are provided for the safe use of the developed tool, in particular, regarding the use of VPN, the creation of separate browser profiles with fake accounts, taking into account the language settings of the system and preliminary testing in the verification mode. The "KotOSINT" application has the potential to become an important tool in combating cybercrime and increasing the effectiveness of OSINT research in the modern information environment, where social media play a key role in communication and dissemination of information. The development of such tools is especially relevant in the context of increasing cyber threats and the need to develop digital intelligence tools.

Keywords: OSINT, social media, browser, user profile, nickname, Selenium.

DOI 10.15276/imms.v15.no2.240
УДК 004.02:517.958

Informatics and Mathematical Methods in Simulation
Vol.15 (2025), No. 2, pp. 240-246

РОЗВ'ЯЗАННЯ АБСТРАКТНОЇ ЗАДАЧІ РІМАНА ЗА МЕТОДОМ ФАКТОРИЗАЦІЇ

О.Л. Комарніцький, Л.М. Колмакова, М.О. Юрченко

Національний університет «Одеська політехніка»
1, Шевченка пр., м. Одеса, 65044, Україна
Emails: komarnitskii@op.edu.ua, kolmakova@op.edu.ua , yurchenko@op.edu.ua

Найбільш ефективними із загального арсеналу методів дослідження фізичних процесів розповсюдження звукових та електромагнітних сигналів в середовищах зі змінними характеристиками та формування мікротріщин у конструкційних матеріалах, які мають різноманітні неоднорідності спадкового походження, сьогодні є методи із застосуванням інтегралів Коші, крайових задач теорії аналітичних функцій, зокрема крайової задачі Рімана, а також метод сингулярних інтегральних рівнянь [1-3]. Теорія абстрактного сингулярного рівняння та абстрактної задачі Рімана виникла на основі теорії сингулярного інтегрального рівняння та крайової задачі Рімана на замкненому контурі. Розглянуті раніше схеми розв'язання абстрактної задачі Рімана узагальнюють не лише крайові задачі Рімана в просторі Гельдера та L_p, але й деякі інтегральні рівняння типу згортки (з двома ядрами, Вінера-Хопфа, парними у просторі $L_2(\mathbb{R})$ та в більш широких просторах узагальнених функцій). У такій постановці відповідна задача Рімана більше не є крайовою задачею для аналітичних функцій. Однак, незважаючи на загальність, матрична крайова задача Рімана на замкненому контурі не підпорядковується розглянутим схемам. У цій роботі для розв'язання задачі Рімана запропоновано абстрактну схему з іншою аксіоматикою, яка усуває цей недолік. Визначена конструктивна схема рішення задачі в припущенні, що її оператор допускає факторизацію. Однак у абстрактній схемі, як і в теорії матричної крайової задачі Рімана, факторизацію оборотного елемента абстрактної алгебри, що розкладається в топологічну пряму суму двох своїх підпросторів, вдається здійснити лише в окремих конкретних випадках. На практиці кожний новий випадок побудови факторизації має наукову цінність як з теоретичної точки зору, так і з точки зору їх можливих застосувань.

Ключові слова: абстрактна задача Рімана, факторизація, частинні індекси, умови розв'язності, логарифм лінійного оператора.

Вступ. Теорія абстрактного сингулярного рівняння та абстрактної задачі Рімана виникла на ґрунті теорії сингулярного інтегрального рівняння та крайової задачі Рімана на замкненому контурі.

Розглянуті раніше схеми розв'язання абстрактної задачі Рімана узагальнюють не лише крайові задачі Рімана у просторі Гельдера та L_p, але й деякі інтегральні рівняння типу згортки (з двома ядрами, Вінера - Хопфа [4], парні [5]) у просторі $L_2(\mathbb{R})$ та в більш широких просторах узагальнених функцій. Відповідна до них задача Рімана вже не є крайовою задачею для аналітичних функцій.

Проте, незважаючи на усю загальність, розглянутим схемам не підкоряється матрична крайова задача Рімана на замкненому контурі. Введемо іншу аксіоматику, яка виправляє цей недолік.

Основна частина. Нехай L – банаховий простір, на якому визначений лінійний обмежений інволютивний оператор S. Цей оператор породжує два проектори

$$P_{\pm}=\frac{1}{2}\left(I\pm S\right),\ I$$ – одиничний оператор.

Простір L розкладається в пряму суму образів цих проекторів:

$$L=L_{+}\oplus L_{-}.$$

Розглянемо задачу про знаходження елементів $\varphi^{+}\in L_{+}$ та $\varphi^{-}\in L_{-}$, які задовольняють умові:

$$\varphi^{+}=A\ \varphi^{-}+g, \tag{1}$$

де A – лінійний обмежений оператор, що має обернений оператор і визначається на L,

g – заданий елемент простору L.

Розглянемо спочатку задачу (1) в частинних випадках, коли оператор А має спеціальний вигляд. Припустимо, що на просторі L задані лінійні обмежені оператори

$$U_1,U_2,\ldots,U_n,$$

які задовольняють наступним умовам:

1) $U_j\cdot U_k=U_k\cdot U_j,\quad j,k=\overline{1,n};$

2) для будь-яких $\varphi^{\pm}\in L_{\pm}$

$$U_j\varphi^{+}\in L_{+},\ \ U_j^{-1}\varphi^{-}\in L_{-},\quad j=\overline{1,n}.$$

3) для оператора U_j існує єдиний з точністю до сталого множника елемент $h_j^{+}\in L_{+}$ такий, що

$$U_j^{-1}h_j^{+}\in L_{-},\quad j=\overline{1,n}.$$

4) $U_jh_k^{+}=h_k^{+}$, якщо $k=\overline{1,n}$ та $j\neq k$.

Отримаємо розв’язок задачі (1) в кожному з чотирьох частинних випадків:

1. $A=U_j,\quad j=\overline{1,n}.$

2. $A=U_j^{\varkappa},\quad j=\overline{1,n},\ \varkappa\in\mathbb{N},\ \varkappa>1.$

3. $A=U_j^{-1},\quad j=\overline{1,n}.$

4. $A=U_j^{-\varkappa},\quad j=\overline{1,n},\ \varkappa\in\mathbb{N}.$

1. $A=U_j,\quad j=\overline{1,n}.$

Тобто треба знайти елементи $\varphi^{\pm}\in L_{\pm}$ по умові:

$$\varphi^{+}=U_j\varphi^{-}+g \tag{2}$$

Представимо елемент g у вигляді

$$g=g^{+}-g^{-},\text{ де }g^{\pm}=\pm P_{\pm}g.$$

Тоді задача (2) буде мати вигляд:

$$\varphi^{+}-g^{+}=U_j\varphi^{-}-g^{-}.$$

Застосуємо до обох частин рівності оператор U_j^{-1}:

$$U_j^{-1}\left(\varphi^+ - g^+\right) = \varphi^- - U_j^{-1}g^-,$$

оскільки $\varphi^- - U_j^{-1}g^- \in L_-$, то й $U_j^{-1}\left(\varphi^+ - g^+\right) \in L_-$,

тоді $\varphi^+ - g^+ = c \cdot h_j^+$, c - довільна стала.

Отримаємо розв'язок задачі (2):

$$\varphi^+ = g^+ + c \cdot h_j^+,\ \varphi^- = U_j^{-1}g^- + c \cdot U_j^{-1}h_j^+.$$

2. $A = U_j^{\varkappa},\ \ j = \overline{1,n},\ \varkappa \in \mathbb{N},\ \varkappa > 1.$

$$\varphi^+ = U_j^{\varkappa}\varphi^- + g, \tag{3}$$

$$\varphi^+ - g^+ = U_j^{\varkappa}\varphi^- - g^-,$$

$$U_j^{-\varkappa}\left(\varphi^+ - g^+\right) = \varphi^- - U_j^{-\varkappa}g^-,$$

$$\varphi^- - U_j^{-\varkappa}g^- \in L_-,$$

$$U_j^{-\varkappa}\left(\varphi^+ - g^+\right) \in L_-.$$

В цьому випадку:

$$\varphi^+ - g^+ = \sum_{k=0}^{\varkappa-1} c_k U_j^k h_j^+,$$

$c_0, c_1, \ldots, c_{\varkappa-1}$ – довільні сталі.

Отримаємо розв'язок задачі (3) у вигляді:

$$\varphi^+ = g^+ + \sum_{k=0}^{\varkappa-1} c_k U_j^k h_j^+,$$

$$\varphi^- = U_j^{-\varkappa}g^- + \sum_{k=0}^{\varkappa-1} c_k U_j^{-\varkappa+k} h_j^+.$$

3. $A = U_j^{-1},\ \ j = \overline{1,n}.$

$$\varphi^+ = U_j^{-1}\varphi^- + g, \tag{4}$$

$$\varphi^+ - g^+ = U_j^{-1}\varphi^- - g^-,$$

елемент у лівій частині $\varphi^+ - g^+ \in L_+$, елемент у правій частині $U_j^{-1}\varphi^- - g^- \in L_-$.

Отже, $\varphi^+ - g^+ = 0$, $U_j^{-1}\varphi^- - g^- = 0$.

Єдиний розв'язок задачі (4) в цьому випадку має вигляд:

$$\varphi^+ = g^+,\ \varphi^- = U_j g^-.$$

Зауважимо, що $U_j g^-$ може не належати L_-, тоді задача (4) в загальному випадку не має розв'язку.

Виявимо, якою буде умова існування розв'язку задачі в цьому випадку.

Простір L_- розкладається в пряму суму двох підпросторів L_-^1 та $U_j^{-1}\ L_-$, L_-^1 – лінійна оболонка, натягнута на елемент $U_j^{-1}h_j^+$.

Таким чином, будь-який елемент $\psi^- \in L_-$ єдиним способом представляється у вигляді $\psi^- = c\cdot U_j^{-1}h_j^+ + U_j^{-1}\psi_1^-$, c - довільна стала, $\psi_1^- \in L_-$.

Розглянемо лінійний функціонал $F_j^{\ 1}$, заданий на просторі L_-, який діє за правилом:

$$F_j^{\ 1}\ \ \psi^- \ = c.$$

Очевидно, що задача (4) буде мати розв'язок, лише коли виконана умова:

$$F_j^{\ 1}\ \ g^- \ = 0.$$

Зауважимо, що простір L_- можна також розкласти в пряму суму підпросторів $L_-^{\varkappa} \oplus U_j^{\varkappa}\ L_-$, де $L_-^{\varkappa}$ - лінійна оболонка, натягнута на елементи $U_j^{-1}h_j^+,\ U_j^{-2}h_j^+,\ldots,U_j^{-\varkappa}h_j^+$. Тобто довільний елемент $\psi^- \in L_-$ єдиним чином записується у вигляді

$$\psi^- = c_1\cdot U_j^{-1}h_j^+ + c_2\cdot U_j^{-2}h_j^+ \ldots + c_{\varkappa}\cdot U_j^{-\varkappa}h_j^+ + U_j^{-\varkappa}\psi_1^-,\ \ \psi_1^- \in L_-,$$

$c_1,\ldots,c_{\varkappa}$ - довільні сталі.

Розглянемо лінійні функціонали $F_j^{\ 1}$, $F_j^{\ 2}$,…,$F_j^{\ \varkappa}$, що задані на просторі L_-, які діють за правилом:

$$F_j^{\ 1}\ \ \psi^- \ = c_1,\ F_j^{\ 2}\ \ \psi^- \ = c_2,\ldots,F_j^{\ \varkappa}\ \ \psi^- \ = c_{\varkappa}.$$

Далі розглянемо наступний випадок.

4. $A = U_j^{-\varkappa}$.

$$\varphi^+ = U_j^{-\varkappa}\varphi^- + g \tag{5}$$

$$\varphi^+ - g^+ = U_j^{-\varkappa}\varphi^- - g^-,\ \ \varphi^+ - g^+ \in L_+,\ U_j^{-\varkappa}\varphi^- - g^- \in L_-.$$

Розв'язок задачі (5) має вигляд:

$$\varphi^+ = g^+,\ \varphi^- = U_j^{\varkappa}g^-.$$

Для того, щоб $\varphi^- \in L_-$, необхідно та достатньо виконання $\varkappa$ - умов існування розв'язку:

$$F_j^{\ k}\ \ g^- \ = 0,\ \ k = \overline{1,\varkappa}.$$

Узагальнимо розглянуті вище випадки:

$$A = U = U_1^{\varkappa_1}U_2^{\varkappa_2}\ldots U_n^{\varkappa_n}.$$

$$\varphi^+ = U\ \varphi^- + g, \tag{6}$$

Припустимо: $\varkappa_j > 0, j = \overline{1,m},\ \varkappa_j < 0, j = \overline{m+1,l}, \varkappa_j = 0,\ \ j = \overline{l+1,n}$.

Для існування розв'язку задачі (6) необхідно та достатньо виконання $-\sum_{j=m+1}^{l}\varkappa_j$ умов:

$$F_j^{\,k}\ \ g^-\ =0,\ \ j=\overline{m+1,l},\ \ k=\overline{1,-\varkappa_j}\,.$$

І якщо ці умови виконані, то загальний розв'язок задачі (6) залежить від $\sum_{j=1}^{m}\varkappa_j$ довільних сталих $c_{jk}\ \ k=\overline{0,\varkappa_j-1},\ j=\overline{1,m}$, та має вигляд:

$$\varphi^+=g^+ +\sum_{j=1}^{m}\sum_{k=0}^{\varkappa_j-1}c_{jk}U_j^k h_j^+,$$

$$\varphi^-=U^{-1}g^- +\sum_{j=1}^{m}\sum_{k=0}^{\varkappa_j-1}c_{jk}U_j^{-\varkappa_j+k}h_j^+.$$

Припустимо, що довільний оператор A, що має обернений оператор, допускає факторизацію, тобто представляється у вигляді:

$$A=X^+U\left(X^-\right)^{-1}, \tag{7}$$

де X^+, X^- – лінійні оператори, що мають обернені оператори та задані на L і відображають відповідно підпростори $L_\pm$ в себе.

Тоді задача (1) приймає вигляд:

$$\varphi^+=X^+U\left(X^-\right)^{-1}\cdot\varphi^- +g,$$

$$\left(X^+\right)^{-1}\varphi^+=U\left(X^-\right)^{-1}\cdot\varphi^- +\left(X^+\right)^{-1}\cdot g.$$

Заміна:

$$\psi^+=\left(X^+\right)^{-1}\cdot\varphi^+,\ \psi^-=\left(X^-\right)^{-1}\cdot\varphi^-,\ \left(X^+\right)^{-1}\cdot g=q.$$

Отримаємо:

$$\psi^+=U\cdot\psi^- +q \tag{8}$$

Таку задачу розглянуто вище.

Для існування її розв'язку необхідно та достатньо виконання $-\sum_{j=m+1}^{l}\varkappa_j$ умов:

$$F_j^{\,k}\ \ q^-\ =0,\ \ j=\overline{m+1,l},\ \ k=\overline{1,-\varkappa_j},\ q^\pm=\pm P_\pm q.$$

Загальний розв'язок задачі (8) залежить від $\sum_{j=1}^{m}\varkappa_j$ довільних сталих та має вигляд:

$$\psi^+=q^+ +\sum_{j=1}^{m}\sum_{k=0}^{\varkappa_j-1}c_{jk}U_j^k h_j^+,$$

$$\psi^-=U^{-1}q^- +\sum_{j=1}^{m}\sum_{k=0}^{\varkappa_j-1}c_{jk}U_j^{-\varkappa_j+k}h_j^+.$$

Тоді:

$$\varphi^+=X^+\left(q^+ +\sum_{j=1}^{m}\sum_{k=0}^{\varkappa_j-1}c_{jk}U_j^k h_j^+\right),$$

$$\varphi^-=X^-\left(U^{-1}q^- +\sum_{j=1}^{m}\sum_{k=0}^{\varkappa_j-1}c_{jk}U_j^{-\varkappa_j+k}h_j^+\right)$$

– розв'язок задачі (1).

Ми припустили, що оператор A допускає факторизацію, і здійснили конструктивне розв'язання задачі, але не встановили способи представлення оператора у вигляді (7).

Далі розглянемо одну з ситуацій, коли факторизацію можна здійснити.

Нехай T є комутативною підалгеброю з одиницею алгебри лінійних обмежених операторів, заданих на L.

Нехай оператори $A,U_j \in T$, $j \in \overline{1,n}$. На алгебрі задана експоненціальна функція

$$\exp V = \sum_{n=0}^{\infty} \frac{V^n}{n!}.$$

Елементи алгебри T, що мають такий вигляд, називаються логарифмованими. Позначимо множину логарифмованих елементів алгебри $\exp(T)$. Ця множина є групою відносно множення. Якщо оператор $W = \exp V \in exp(T)$, то оператор V називається логарифмом оператора W: $V = \ln W$.

Припустимо, що існує такий набір частинних індексів $\varkappa_1, \varkappa_2, \ldots \varkappa_n$, що оператор $U^{-1} \cdot A \in \exp(T)$, тоді оператор A допускає факторизацію.

В алгебрі T розглянемо задачу по стрибку:

$$\Psi^+ - \Psi^- = \ln U^{-1} \cdot A .$$

Позначимо $X^{\pm} = \exp \Psi^{\pm}$. Елементи $X^{\pm}$ мають обернені відповідно в алгебрах $T_{\pm}$ - операторів, що переводять відповідно простори $L_{\pm}$ в себе, причому має місце факторизація (7).

Висновки. У статті запропонована абстрактна схема розв'язання задачі Рімана, аксіоматіка якої на відміну від запропонованих раніше абстрактних схем, узагальнює матричну крайову задачу Рімана на замкненому контурі l [6]:

$$\Phi^+(t) = G(t)\Phi^-(t) + g(t),\ t \in l.$$

Тут $G(t)$ – матриця гельдеровских функцій, яка має обернену матрицю, $g(t) \in H_{\lambda}$, - стовпець гельдеровских функцій, $\Phi^+(t)$ складається з функцій, що аналітично продовжуються в середину контуру, $\Phi^-(t)$ – поза контуру, S – оператор сингулярного інтегрування, точка 0 належить області, що обмежена контуром l.

Оператором U_j в цьому випадку є діагональна квадратна матриця, у якої j-ий елемент діагоналі дорівнює t, а усі інші діагональні елементи дорівнюють 1. Елементом h_j^+ є стовпець, в якому в j-му рядку міститься стала функція, тотожно рівна 1.

Очевидно, що оператор U представляє діагональну матрицю з функціями $t^{\varkappa_1}, t^{\varkappa_2}, \ldots, t^{\varkappa_n}$ на діагоналі.

Список літератури

1. Oborskij G. A., Dashhenko A.F., Usov A. V., Dmitrishin A. V. Modelirovanie sistem : Monografija, Odesa: Astroprint. 2013. 664 p
2. Usov A.V., Kunitcyn M.V. Simulation of thermomechanical processes in functionally-gradient materials of inhomogeneous structure in the manufactturingand operation of rocket structural elements. Kosm.tex. Raket. vooruz. Space Technology. Missile Armaments. 2020(1). C. 137-148. https://doi.org/10.33136/stma2020.01.137.

3. Kryvyi O.F., Morozov Yu.O. Influence of concentrated forces on an interface inclusion under the conditions of smooth contact in the inhomogeneous transversely isotropic space, J.Math.Sci., Vol. 279, No 2. 2024. C. 197–212. https://doi.org/10.1007/s10958-024-07005-3.
4. Комарницький О.Л., Колмакова Л.М. Метод факторизации при решении двохиндексной системи Винера–Хопфа. Міжнародна літня математична школа пам'яті В.А. Плотникова. 15-22 червня 2013 р. Одеса. Україна. Тези доповідей. С. 68.
5. Комарницький О.Л., Колмакова Л.М. Метод факторизації при рішенні парної двохіндексної системи типу згортки. Міжнародна літня математична школа пам'яті В.А. Плотникова. 11-16 червня 2018 р. Одеса. Україна. Тези доповідей. С.59.
6. Комарницький О.Л., Колмакова Л.М. Абстрактна задача Рімана, яка узагальнює матричну крайову задачу. Матеріали XVII Міжнародної наукової конференції імені академіка Михайла Кравчука, Том 1, Київ, 19-20 травня 2016 р. С. 150–151.

THE ABSTRACT RIEMANN PROBLEM AND THE FACTORIZATION METOD

O.L. Komarnitskii, L.M. Kolmakova, M.O. Yurchenko

National Odesa Polytechnic University
1, Shevchenko Ave., Odesa, 65044, Ukraine
Emails: komarnitskii@op.edu.ua, kolmakova@op.edu.ua, yurchenko@op.edu.ua

The most effective among the general arsenal of methods for studying the physical processes of propagation of acoustic and electromagnetic signals in media with variable characteristics, as well as the formation of microcracks in structural materials that have various types of hereditary inhomogeneities, at present are methods involving Cauchy integrals, boundary value problems of the theory of analytic functions , in particular the Riemann boundary value problem, as well as the method of singular integral equations. The theory of abstract singular equation and of the abstract Riemann problem arose on the basis of the theory of a singular integral equation and the Riemann boundary value problem on a closed contour. The previously considered schemes for solving the abstract Riemann problem generalize not only the Riemann boundary value problems in Holder space and L_p, but and some integral convolution type equations (with two kernels, Wiener–Hopf, pair ones) in the space $L_2(\mathbb{R})$ and in wider spaces generalized functions. The corresponding Riemann problem is no longer a boundary problem for analytic functions. However, despite on the whole generality, the matrix Riemann boundary-value problem on a closed contour does not obey to the considered schemes. In this paper, for solving the Riemann problem, an abstact scheme with another axiomatic is proposed, which eliminates this disadvantage. A constructive scheme for solving the problem has been defined under the assumption that its operator permits factorization. However, in the abstract scheme, as in the theory of the matrix Riemann boundary value problem, the factorization of an invertible element of an abstract algebra, decomposing into a typological direct sum of its two subspaces, it is possible to build it only in special cases. In practice, each new case of constructing a factorization is of scientific value both from a theoretical perspective and in terms of its potential applications

Keywords: the Riemann abstract problem, factorization, partial indices, solvability conditions, logarithm of linear operator.

DOI 10.15276/imms.v15.no2.247
УДК 004.056.55

Informatics and Mathematical Methods in Simulation
Vol.15 (2025), No. 2, pp. 247-259

АЛГОРИТМ ВИЯВЛЕННЯ ФІШИНГУ В ЛИСТАХ МЕСЕНДЖЕРІВ ТА ЕЛЕКТРОННОЇ ПОШТИ ІЗ ВИКОРИСТАННЯМ НЕЙРОННИХ МЕРЕЖ З ФІКСОВАНОЮ КІЛЬКІСТЮ РАНЖОВАНИХ ЧИННИКІВ РИЗИКУ

В.О. Назаров, А.В. Садченко, О.А. Кушніренко

Національний університет «Одеська політехніка»
1, Шевченка пр., м.Одеса, 65044, Україна
Emails: koa@op.edu.ua, sadchenko@op.edu.ua

Стаття присвячена розробці алгоритмів виявлення найбільш шкідливого різновиду спаму – фішингу на основі нейронних мереж із сигмоїдною та пороговою функціями активації. Проведено огляд та аналіз існуючих підходів до виявлення спаму і встановлено, що в більшості сучасних алгоритмів використовується кількісний підхід заснований на постійному поповненню словників словосполучень у явному та токенізованому вигляді, що при використанні, наприклад, нейромережі на етапі прийняття рішення може збільшувати імовірність віднесення етичного змісту повідомлень та листів до спаму. Представлено новий підхід до формування вектору вхідних чинників, що можуть бути присутні як у спамі так і в етичному контенті. Було виділено чотири основних загальних чинника, що ранжуються по десятибальної шкалі ризику, це «можлива винагорода», «обмеження у часі», «емоційне забарвлення» та «наявність посилання на зовнішній ресурс». Щодо перших 3-х чинників запропонована 10-ти бальна шкала оцінки впливу згідно аналізу наявності шаблонів-словосполучень, а «наявність посилання на зовнішній ресурс» має дві градації: «1», якщо посилання є, та «0» якщо його немає. Таке ранжування дозволило створити цифрове уявлення факторів ризику у вигляді вхідного вектору чинників, що потім подається на вхід нейромережі із сигмоїдною чи пороговою функціями активації. Нейромережа із використанням сигмоїдної функції активації після навчання, на основі десяти випадків фішинг контенту та такої же кількості етичних листів, забезпечила імовірність виявлення шкідливого контенту приблизно 10^{-3} при умові віднесення підозрілих листів до категорії спам. Нейромережа із пороговою функцією активації і додатковим розбиттям діапазону чинника «наявність посилання на зовнішній ресурс» на десять умовних градації, за допомогою аналізу S матриці взаємного впливу, показала високу швидкість навчання та імовірність виявлення шкідливого контенту приблизно $0.5 \cdot 10^{-3}$. Результати виконаних досліджень можна використовувати при створені фішингових фільтрів як із жорстким, так і з м'яким рішенням, що призначені для вбудовування в поштові сервіси та месенджери соціальних мереж.

Ключові слова: фільтри спаму, фішингові листи, нейромережа із сигмоїдною функцією активації, чинники ризику.

Вступ. В сучасних реаліях дистанційна освіта та робота набувають вимушеної популярності. І якщо ще зовсім недавно основним каналом дистанційного обміну інформації була електронна пошта [1], то на сьогоднішній день оперативний обмін інформацією здійснюється через численні месенджери, такі як Viber, Telegram, WhatsApp. При цьому поштові сервіси, наприклад, від Google містять потужні інструменти щодо боротьби зі шкідливими чи спам-листами. Спам-фільтри зазвичай інтегровані до поштових сервісів і побудовані із використанням технологій на базі штучного інтелекту та нейронних мереж [2,3]. У таких умовах для успішної, з погляду зловмисника, доставки шкідливої інформації необхідно застосовувати дедалі складніші методи обходу [4,5] спам-захисту. Наприклад, адреса відправника може бути підроблена під адресу корпоративної пошти [6].

На відміну від самонавчених технологій поштових сервісів, месенджери в більшості випадків такого захисту не мають або використовують пропрієтарні

алгоритми з нерегламентованим ступенем захисту. Тому актуальним є завдання пошуку нових підходів до виявлення шкідливої інформації.

Під шкідливою інформацією чи спам-повідомленнями будемо мати на увазі такі повідомлення [7], що надіслані великій групі одержувачів без їх попередньої згоди.

Серед спам-листів є відносно безпечні, такі що використовуються для реклами товарів чи послуг. Більшою небезпекою володіють листи чи повідомлення що містять посилання [8] на веб-сайти зловмисного програмного забезпечення або фішингового хостингу, через які може бути вкрадено приватна інформація чи безпосередньо гроші. При розповсюдженні небезпечних повідомлень за допомогою месенджерів може використовуватись ID відправника, що добре відомий отримувачу.

Розглянемо основні підходи до реалізації спам-фільтрів.

Спам-фільтри на основі списків. Метод боротьби зі спамом за допомогою спам-фільтрів на основі списків [9,10] передбачає використання:

– чорних списків (blacklists) – список адрес/доменів, з яких ніколи не слід приймати повідомлення;

– білих списків (whitelists) – список надійних відправників, чиї листи завжди пропускаються;

– сірих списків (greylists) – тимчасово відхиляються листи від невідомих відправників, з можливістю їх повторної доставки пізніше.

Переваги спам-фільтрів на основі списків:

– простота реалізації та легкість налаштування, зрозумілість у використанні;

– висока ефективність проти відомих джерел спаму – якщо відправник вже у чорному списку, його листи будуть заблоковані;

– низькі обчислювальні витрати що не вимагають складного аналізу тексту або машинного навчання;

– мінімальна ймовірність помилкових спрацьовувань щодо білих списків та гарантована доставка від надійних відправників.

Недоліки спам-фільтрів на основі списків:

– обмежена гнучкість – не здатні справлятися з новим або змінним спамом (невідомі джерела);

– високий ризик хибно позитивних спрацьовувань – легітимний відправник може опинитися в чорному списку помилково;

– залежність від оновлення списків – потрібне постійне оновлення для збереження актуальності;

– можна обійти – спамери легко змінюють адреси або використовують зламані сервери, щоб не потрапити до списку;

– сірі списки можуть викликати затримки доставки, особливо для нових відправників.

Байєсівські фільтри. Принцип дії байєсовського фільтру [11-16] заснований на статистичному аналізі тексту листа на основі ймовірності того, що він містить спам, тобто використовується модель навчання із вчителем.

Серед переваг байєсовських фільтрів можна виділити можливість самонавчання, ефективність, особливо при великому обсязі попередньо оброблених даних. Такі фільтри здатні виявляти новий спам, навіть із невідомих адрес.

До недоліків байєсовських фільтрів можна віднести наступне:

– потрібність попереднього навчання на великій вибірці спаму та нормальних листів;

– складність визначення стану навчання завдяки чому можлива помилкова ідентифікація шкідливого контексту;

– можливе різке погіршення якості роботи фільтру при використанні методів обфускації (маскування) тексту [17].

Евристичні фільтри. Принцип дії евристичного фільтру [18] полягає в формуванні спеціалізованого набору правил і шаблонів щодо змісту листів, наприклад, наявність підозрілих посилань, певних слів, формату листа.

До переваг евристичних фільтрів можна віднести:

– гнучкість завдяки доброму виявлянню шаблонного спаму;

– відсутність залежності від конкретного відправника.

Недоліки евристичних фільтрів:

– можуть давати помилкові спрацьовування при відсутності відповідного шаблону;

– вимагають постійного поповнення словнику ключових слів;

– важко підтримувати універсальність правил.

Використання нейромережі. Переваги нейромереж (НМ) [19-21] у виявленні спаму:

– краще відношення вірно виявлених випадків спаму до загального числу проаналізованих повідомлень порівняно із іншими методами;

– нейромережі можуть навчатися з нових даних і адаптивно підлаштовуватися під зміну тактик зловмисників;

– нейромережі можна використовувати щодо виявлення складних шаблонів текстів, на відміну від простих фільтрів, наприклад, на основі аналізу ключових слів, нейромережі виявляють складні взаємозв’язки між словами, стилем, структурою повідомлень тощо.

До недоліків існуючих алгоритмів виявлення спаму із використанням нейромережі можна віднести:

– необхідність великої кількості заздалегідь підготовлених даних, що мають властивість у вигляді спам чи не спам;

– високі вимоги до обчислювальних ресурсів, таких як навчання та обробка великих обсягів даних, наявності потужних процесорів чи графічних карт та оперативної пам’яті;

– наявність ефекту «Чорна скринька» – іноді важко зрозуміти, чому мережа класифікувала лист як спам;

– наявність ефекту «Перенавчання» – якщо не ввести обмеження на кількість листів щодо навчання, може відбутись поступове погіршення характеристик виявлення шкідливого контенту;

– уразливість до атак – нейромережі можуть бути обмануті (наприклад, додаванням "етичних" словосполучень у лист, щоб обійти фільтр).

Загальним для вище перелічених методів виявлення шкідливого контенту в повідомленнях є відсутність критеріїв щодо ранжування рівня загрози [22-27], що виходить від окремих виявлених чинників та взаємного впливу дієвих чинників листу на його кінцеве призначення.

Метою даної статті є розробка алгоритму виявлення фішинг-спаму на основі одношарової нейромережі, та методу ранжування рівнів загрози, що виходить від спільних чинників щодо етичних та спам листів.

Основна частина. Розглянемо основні етапи аналізу листів з метою виявлення шкідливого контенту, рис 1.

Одним з найскладніших етапів даного алгоритму є виявлення шкідливих рис вмісту листів, що аналізуються, чи чинників впливу, ранжування по ступеню риску та перетворення у числову форму для формування вхідного вектору нейромережі.

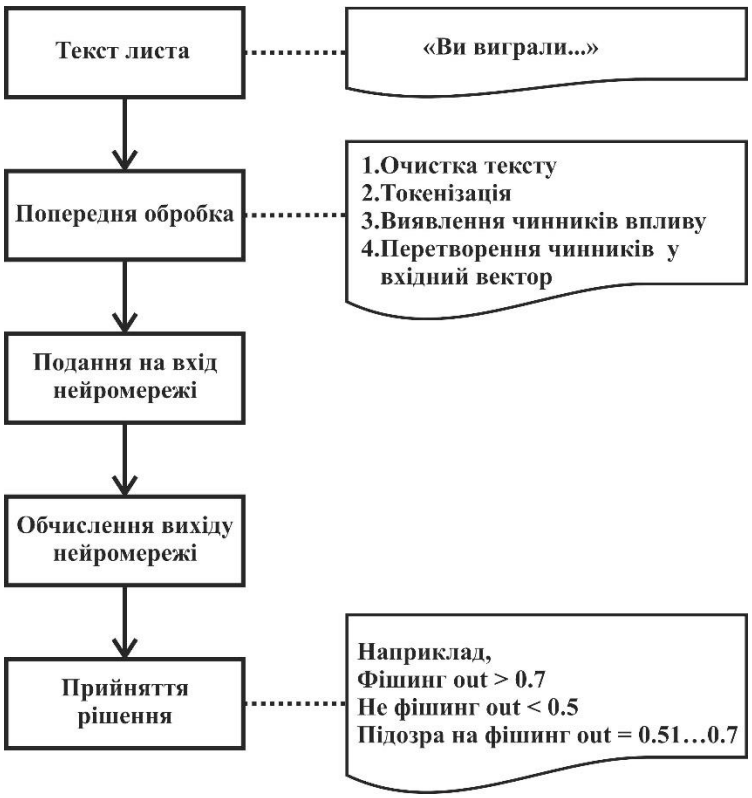


Рис. 1. Загальна структура алгоритму побудови антіспам-фільтру

Для виявлення найбільш впливових чинників ризику було проаналізовано зміст найбільш поширених фішингових листів, що стосуються отримання грантів, виграшу у лотереї та банківської діяльності.

Загальні риси шкідливих листів. Аналіз найбільш шкідливих спам-листів, тобто таких, що можуть призвести до фінансових та іміджевих втрат дозволяє виявити основні чинники, що можуть використовуватись для здійснення маніпуляції:

- психологічний тиск;
- шантаж або маніпуляція;
- запити на конфіденційні дані;
- підроблені або масковані посилання;
- маніпулятивна мова;
- відсутність чіткої ідентифікації відправника;
- відсутність альтернатив зв’язку;
- обмеження у часі;
- обіцянки великих виграшів або ексклюзивних можливостей;
- емоційне забарвлення та перебільшення.

Слід зазначити, що листи які уявляють із себе спам-фішинг завжди містять посилання на зовнішній шкідливий ресурс.

Однак е загальні чинники, що можуть міститись як в етичних, так і в спам-листах:

- запити на конфіденційні дані;
- відсутність чіткої ідентифікації відправника;
- відсутність альтернатив зв’язку;
- обмеження у часі;
- обіцянки великих виграшів чи винагорода за виконану роботу;
- емоційне забарвлення та перебільшення.

З метою створення математичної моделі алгоритму виявлення спаму пропонується створити *вектор чинників*, що є загальними для етичних та спам-листів.

Визначення балів щодо окремих складових можливе завдяки використанню нейромережі з метою періодичного оновлення вектору чинників.

Виявимо чинники, що можливо уявити у вигляді числа балів. Наприклад, деякі медичні застосунки пропонують ліки в залежності від суб'єктивної оцінки самопочуття в балах.

Розглянемо декілька текстів спам-листів, що містять вказівку суми нагороди у явному вигляді.

Спам-лист, що містить інформацію про фіктивний грантовий проект: «Ваша успішність перевершує очікування, і Вам надана можливість отримати грантову підтримку у розмірі 100 000 USD для розвитку Ваших проектів».

Спам-лист, що містить інформацію про виграш в лотерею: «Ми з величезним задоволенням повідомляємо, що Ваша участь у престижній лотереї Royal Fortune Jackpot була успішною, і Ви стаєте щасливим переможцем головного призу у розмірі 75 000 USD!»

Ще варіант спам-листу, що містить інформацію про виграш в лотерею:

«Це надзвичайно важливе повідомлення! Ми з великим захопленням і водночас із глибокою тривогою повідомляємо, що результати останньої лотереї Platinum Prize Bonanza підтвердили Ваш виграш у розмірі 50 000 USD!»

Позначимо параметр миттєвої винагороди чи плати за виконану роботу як R_{m}. Максимальний запропонований виграш згідно проаналізованих спам-листів складає 100 000 USD, тому, щоб отримати оцінку R_{m} по 10-ти бальної шкалі:

$$R_m, балів = \begin{cases} 10, & R_m > 100000 \\ 2\left[Log_{10} R_m\right], & R_m < 100000 \end{cases},$$

де $\left[X\right]$ – ціла частина числа.

Параметр обмеження у часі T_{m} також приведемо до системи оцінювання за 10-ти бальної шкалою.

Приклад відповідності змісту листів чиннику обмеження у часі приведений в таблиці 1. Відсутність обмежень означає що $T_{\text{m}} = 0$.

Таблиця 1.

Відповідність змісту листів чиннику обмеження у часі T_{m}

Зміст листа	$\boldsymbol{T_{\text{m}}}$**, балів**
обмеження відсутні	0
«…до визначеної дати»	1
«…протягом двох тижнів»	2
«…протягом тижня»	3
«…бажано до післязавтра»	4
«…бажано до завтра.»	5
«…протягом наступних 24 годин»	6
«…протягом наступних 12 годин»	7
«Ви маєте лише 6 годин»	8
«…будь-яка затримка»	9
«Вам необхідно негайно»	10

Параметр емоційного забарвлення та перебільшення позначимо E_{z}.

Приклад відповідності змісту листів чиннику емоційного забарвлення приведений в таблиці 2. Відсутність забарвлення означає що $E_z = 0$.

Таблиця 2.

Відповідність змісту листів чиннику емоційного забарвлення

Зміст листа	E_z**, балів**
відсутність забарвлення	0
оплата виконаної роботи гарантована	1
ми цінуємо Ваш час тому буде надано авансовий платіж	2
при вчасному виконанні роботи розмір винагороди буде збільшено	3
Вам надана можливість отримати грантову підтримку	4
Вам нараховано бонуси	5
Ваша успішність перевершує очікування	6
розмір винагороди було збільшено	7
Ви претендуєте на отримання премії	8
Ви стаєте щасливим переможцем головного призу	9
нагороду вже надіслано	10

Також введемо чинник ризику R_z, що буде відповідати за імовірність хибної ідентифікації етичного листа як спам-листу. Також цей чинник необхідно прив'язати до факту наявності посилання на зовнішній шкідливий ресурс, наприклад якщо посилання нема то початкове значення $R_z = 0$, а якщо посилання є то $R_z = R_{z\,max}$.

Отримаємо наступний загальний вигляд вектору чинників, що містяться в листі чи повідомленні:

$$V_p = \left[R_m, T_m, E_z, R_z\right],$$

де R_m – чинник винагороди;

T_m – чинник обмеження у часі;

E_z – чинник емоційного забарвлення;

R_z – чинник ризику.

Опишемо алгоритм визначення спам-листів із використанням одношарової нейромережі з використанням нейрону з найбільш поширеною – сигмоїдною функцією активації та критерієм середньоквадратичного відхилення щодо отримання помилки. Архітектура нейрону приведена рис 2.

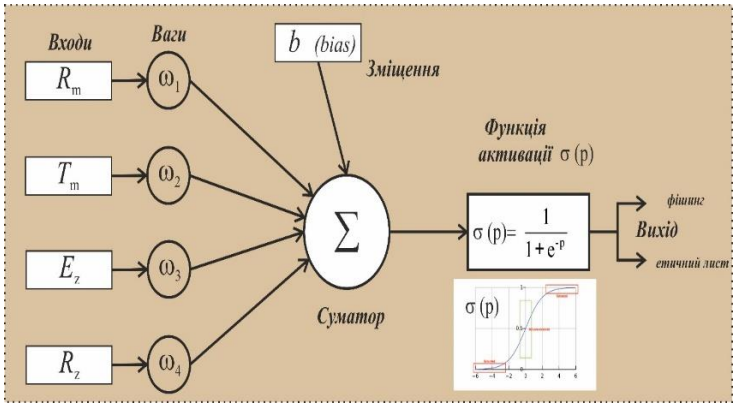


Рис. 2. Нейрон з 4-мя входами та сигмоїдною функцією активації

Крок 1. Обчислюємо значення аргументу p функції активації із використанням виразу:

$$p = V_p{}^T W + b = \sum_{i=1}^{4} v_i w_i + b = R_m w_1 + T_m w_2 + E_z w_3 + R_z w_4 + b,$$

де ω_1, ω_2, ω_3, ω_4 – це вагові коефіцієнти,

b – значення параметру початкового зсуву (bias).

Обчислюємо вихідне значення нейрону:

$$y = \sigma(p) = \frac{1}{1+e^{-p}}$$

Крок 2. Обчислюємо значення похибки чи параметр втрат L як середньоквадратичне відхилення (MSE) отриманого значення y від цільового значення y_a:

$$L = \frac{1}{2}(y - y_a)^2$$

Крок 3. Знайдемо похідну функції помилки з використанням градієнтного методу чи методу зворотного розповсюдження помилки:

$$\frac{dL}{dw_i} = \frac{dL}{dy} \cdot \frac{dy}{dp} \cdot \frac{dp}{dw_i},\ i = \overline{1,4}$$

Похідна помилки по вихідному значенню нейрона:

$$\frac{dL}{dy} = \frac{d}{dy}\left(\frac{1}{2}(y - y_a)^2\right) = 2 \cdot \frac{1}{2}(y - y_a) = y - y_a$$

Похідна вихідного значення по аргументу функції активації нейрона:

$$\frac{dy}{dp} = \frac{d}{dp}\left(\frac{1}{1+e^{-p}}\right) = -\frac{\left(1+e^{-p}\right)_p'}{\left(1+e^{-p}\right)^2} = -\frac{(1)_p' + (-z)_p' \cdot e^{-z}}{\left(1+e^{-p}\right)^2} = -\frac{0 + (-1 \cdot (z)_p') \cdot e^{-p}}{\left(1+e^{-p}\right)^2} = \frac{e^{-p}}{\left(1+e^{-p}\right)^2}$$

Похідна помилки щодо кожної ваги та параметра b:

$$\frac{dL}{dw_i} = \frac{dL}{dp} \cdot v_i,\ i = \overline{1,4}$$

$$\frac{dL}{db} = \frac{dL}{dp} \cdot b,$$

де $\frac{dL}{dp} = \frac{dL}{dy}\frac{dy}{dp}$

Крок 4. Виконуємо оновлення ваг при заданому параметрі швидкості навчання η, що може приймати значення, наприклад, $\eta = 0.1$:

$$w_i^{new} = w_i - \eta \cdot \frac{dL}{dw_i}, i = \overline{1,4}$$

$$b^{new} = b - \eta \cdot \frac{dL}{db},$$

де $\omega_i{}^{new}$ b^{new} – оновлені значення вагових коефіцієнтів на початкового зсуву.

Крок 5. Процес навчання у вигляді повторення кроків 1…4 повторюється щодо множини вхідних векторів поки не буде отримане задане значення помилки.

Перевіримо роботу алгоритму в режимі навчання.

Результати роботи нейромережі в режимі навчання при початкових параметрах (швидкість навчання η=0.1, вагові коефіцієнти $\omega_i = 0.1$, $b = 0.5$) наведені в таблиці 3.

Таблиця 3.

Процес навчання нейромережі

E_m	T_m	R_z	E_z	y	y_a	**Помилка «є» чи «ні»**
0	0	0	0	0.06	0	ні
0	0	0	1	0.03	0	ні
1	1	1	0	0.15	0	ні
1	1	1	1	0.09	0	ні
2	2	2	0	0.33	0	ні
2	2	2	1	0.21	0	ні
3	3	3	0	0.59	0	Є
3	3	4	0	0.44	0	ні
4	3	4	1	0.49	0	ні
4	4	4	0	0.53	1	ні
4	4	4	1	0.80	1	ні
5	5	5	0	0.69	1	ні
5	5	5	1	0.91	1	ні
6	6	6	0	0.86	1	ні
6	6	6	1	0.96	1	ні
7	7	7	0	0.94	1	ні
7	7	7	1	0.98	1	ні
8	8	8	1	0.98	1	ні
9	9	9	1	0.99	1	ні

Графік залежності виходу нейрона від кількості навчальних векторів показаний на рисунку 3.

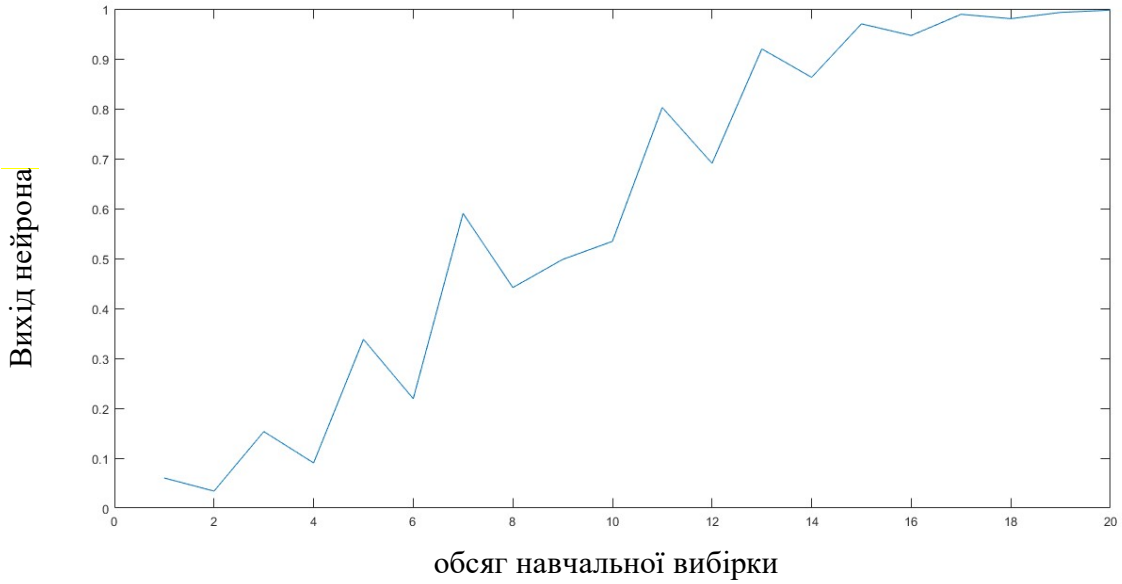


Рис. 3. Залежність виходу нейрона від кількості навчальних векторів

Як можна побачити, помилки ідентифікації спам-контенту відбуваються тільки на ранніх етапах навчання, тобто при недостатньої кількості вхідних навчальних векторів. Можна зробити висновок, що при кількості навчальних векторів більше 10, імовірність помилки різко зменшується і даний алгоритм можна використовувати.

Розглянемо ще одну модифікацію алгоритму виявлення спаму із навчанням за мінімальною довжиною навчальної вибірки.

Алгоритм виявлення фішингу із використанням нейрону із пороговою функцією активації.

Крок 1. На основі аналізу тексту повідомлення формуємо вектор вхідних чинників

$$V_p = \left[R_m, T_m, E_z, R_z\right]$$

де R_m, T_m, E_z можуть приймати дискретні значення з діапазону від 0 до 10,

R_z – 0 чи 10 в залежності від наявності чи відсутності посилань на зовнішні джерела.

Крок 2. Задаймо всі вагові коефіцієнті виконавчого нейрона однаковими $\omega_1 = \omega_2 = \omega_3 = \omega_4 = 10$ і формуємо ваговий вектор, тобто $W = \left[10,10,10,10\right]$.

Крок 3. Обчислюємо значення аргументу p функції активації нейрона:

$$p = \frac{1}{\left(\sum_{i=1}^{n} w_i^2\right)} V_p^T W = \frac{\sum_{i=1}^{n} v_i w_i}{\sum_{i=1}^{n} w_i^2},$$

де v_i – елементи вектору V_p,

ω_i – елементи вектору w,

n – кількість врахованих чинників, в даному випадку n = 4.

Крок 4. Обчислюємо значення вихідного сигналу виконавчого нейрону, тобто значення функції активації $F_A(p)$ в залежності від значення аргументу p.

У якості функції активації обираємо порогову функцію:

$$F_A(p) = \begin{cases} 1, & p \geq 0.5 \\ 0, & p < 0.5 \end{cases}$$

Якщо отримано значення «1» то текст листа містить фішинг, а якщо «0» – то цей лист етичний.

Крок 5. Якщо був активований режим навчання то переходимо на крок 6, інакше алгоритм завершуємо.

Крок 6. Перевіряємо чи вірно виявлена наявність чи відсутність фішингу? Якщо вірно процес навчання завершуємо, інакше переходимо на крок 7.

Крок 7. Обчислюємо величину похибки:

$$E = \left|0.5 - p\right| = \left|0.5 - \frac{\sum_{i=1}^{4} v_i w_i}{\sum_{i=1}^{4} w_i^2}\right|$$

Крок 8. Розраховуємо нове значення чинника R_z, R_{zc}

Якщо не було виявлено фішинг, тобто хибно прийнято рішення про відсутність фішингу:

$$R_{zc} = R_{zp} + \left|0.5 - p\right| \frac{1}{\max(R_m, T_m, E_z)} \sum_{i=1}^{4} w_i^2 = R_{zp} + E \frac{1}{\max(R_m, T_m, E_z)} \sum_{i=1}^{4} w_i^2,$$

де R_{zp} – попереднє значення чинника R_z;

max (R_m, T_m, E_z) – максимальне значення по всім чинникам.

Якщо було хибно виявлено фішинг, при умові його відсутності:

$$R_{zc} = R_{zp} - \left|0.5 - p\right| \frac{1}{\max(R_m, T_m, E_z)} \sum_{i=1}^{4} w_i^2 = R_{zp} - E \frac{1}{\max(R_m, T_m, E_z)} \sum_{i=1}^{4} w_i^2,$$

Крок 9. Формуємо новий вектор вхідних даних $V_p = [R_m, T_m, E_z, R_{zc}]$ і переходимо на крок 2.

Перевіримо роботу алгоритму.

Хай наявний лист володіє наступними чинниками: $R_m = 5$, $T_m = 7$, $E_z = 3$, $R_z = 10$.

Крок 1. Формуємо вектор вхідних чинників:

$$V_p = [5, 7, 3, 10].$$

Крок 2. Задаймо всі вагові коефіцієнті нейрона однаковими $\omega_1 = \omega_2 = \omega_3 = \omega_4 = 10$ і формуємо ваговий вектор, тобто $W = [10, 10, 10, 10]$.

Крок 3. Обчислюємо значення аргументу p функції активації нейрону:

$$p = \frac{1}{\left(\sum_{i=1}^{4} w_i^2\right)} V_p^T W = \sum_{i=1}^{4} v_i w_i = \frac{50 + 70 + 30 + 100}{400} = 0.625$$

Крок 4. Обчислюємо значення функції активації: $F_A(0.625) = 1$.

Робимо вірний висновок про те, що текст повідомлення є спам-фішингом.

Сформуємо матрицю S_p взаємних впливів вхідних чинників листу, що аналізується за наступним правилом:

$$S_p = V_p^T * V_p = \begin{bmatrix} R_m \\ T_m \\ E_z \\ R_z \end{bmatrix} * [R_m, T_m, E_z, R_z] = \begin{bmatrix} R_m^2 & R_m T_m & R_m E_z & R_m R_z \\ T_m R_m & T_m^2 & T_m E_z & T_m R_z \\ E_z R_m & E_z T_m & E_z^2 & E_z R_z \\ R_z R_m & R_z T_m & R_z E_z & R_z^2 \end{bmatrix}$$

Елементи матриці S_p – s_{ij} мають фізичний зміст взаємного впливу чинників *i* та *j* один на одного.

Побудуємо матрицю S_p для вищерозглянутого прикладу:

$$S_p = \begin{bmatrix} 5 \\ 7 \\ 3 \\ 10 \end{bmatrix} * [5, 7, 3, 10] = \begin{bmatrix} 25 & 35 & 15 & 50 \\ 35 & 49 & 21 & 70 \\ 15 & 21 & 9 & 30 \\ 50 & 70 & 30 & 100 \end{bmatrix}$$

Максимальне значення елементів, що не знаходіться на головної діагоналі $s_{24} = s_{42}$, тобто чинники часу та ризику мають найбільший вплив один на одного.

Розглянемо тепер етичний лист, текст якого має наступні чинники:

$R_m = 1$– «оплата виконаної роботи гарантована»;

$T_m = 1$ – «…до визначеної дати»;

$E_z = 4$ – «Вам надана можливість отримати грантову підтримку».

Сформуємо вектор чинників V_e вважаючи $R_z = 0$, тобто $V_e = [6, 2, 4, 0]$.

Знайдемо значення аргументу функції активації нейрону:

$$p = \frac{60 + 20 + 40}{400} = 0.3$$

Значення функції активації: F(p)= F(0.3)=0. Таким чином спам чи фішинг вірно не виявлено, лист є етичним.

Максимальне значення елементів, що не знаходіться на головної діагоналі $s_{13} = s_{31}$, тобто чинники винагороди чи плати за виконану роботу та емоційного забарвлення мають найбільший вплив один на одного в більшості етичних листів.

Побудуємо матрицю S_p для чинників етичного листу:

$$S_p = \begin{bmatrix} 6 \\ 2 \\ 4 \\ 0 \end{bmatrix} * \begin{bmatrix} 6 & 2 & 4 & 0 \end{bmatrix} = \begin{bmatrix} 36 & 12 & 24 & 0 \\ 12 & 4 & 8 & 0 \\ 24 & 8 & 16 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Висновки. В даної роботі було проаналізовано найбільш агресивні види спаму – фішингу, що можуть призвести до значних фінансових втрат. Було виявлено, що спам-листи та етичні листи схожої спрямованості, наприклад, грантові пропозиції чи виграши в лотерею мають спільні чинники, але їхнє співвідношення та вплив на кінцевий результат відрізняються. Було виділено такі загальні чинники як «можлива винагорода», «обмеження у часі», «емоційне забарвлення» та «наявність посилання на зовнішній ресурс». Щодо перших 3-х чинників запропонована 10-ти бальна шкала оцінки впливу згідно аналізу наявності шаблонів-словосполучень, а «наявність посилання на зовнішній ресурс» має дві градації 1, якщо посилання є та 0 якщо його немає. Сформований вектор чинників, що можуть вказувати на наявність фішингу був оброблений за допомогою двох алгоритмів на базі одношарової нейромережі – із класичною сигмоїдною функцією активації та із пороговою функцією активації. Нейромережа із сигмоїдною функцією активації показує більшу точність та дозволяє задавати діапазон невизначеності, що можна класифікувати як «підозра на фішинг», однак потребує суттєвого збільшення обсягу навчаючої вибірки. Мінімальний обсяг навчальної вибірки для забезпечення імовірності помилки не менш 10^{-3} – по десять листів фішингових та етичних. Проте нейромережа із пороговою функцію значне простіша в реалізації. Запропоновані алгоритми можна удосконалювати при збільшенні чинників впливу, що підлягають токенізації та ранжуванню.

Список літератури

1. G. V. Cormack, *Email spam filtering: A systematic review*, vol. 1, no. 4. 2006.
2. E. G. Dada, J. S. Bassi, H. Chiroma, S. M. Abdulhamid, A. O. Adetunmbi, and O. E. Ajibuwa,“Machine learning for email spam filtering: review, approaches and open research problems,” *Heliyon*, vol. 5, no. 6, 2019, doi: 10.1016/j.heliyon.2019.e01802.
3. D. Gaurav, S. M. Tiwari, A. Goyal, N. Gandhi, and A. Abraham, “Machine intelligence-based algorithms for spam filtering on document labeling,” *Soft Comput*., vol. 24, no. 13, pp. 9625–9638, 2020, doi: 10.1007/s00500-019-04473-7.
4. N. Kumar, S. Sonowal, and Nishant, “Email Spam Detection Using Machine Learning Algorithms,”*Proc. 2nd Int. Conf. Inven. Res. Comput. Appl. ICIRCA 2020*, no. July 2020, pp. 108–113, 2020, doi:10.1109/ICIRCA48905.2020.9183098.
5. I. AbdulNabi and Q. Yaseen, “Spam email detection using deep learning techniques,” *Procedia Comput. Sci.,* vol. 184, no. 2019, pp. 853–858, 2021, doi: 10.1016/j.procs.2021.03.107.
6. S. M. Yasin and I. H. Azmi, “Email Spam Filtering Technique: Challenges and Solutions,” *J. Theor. Appl. Inf. Technol.,* vol. 101, no. 13, pp. 5130–5138, 2023
7. A. S. Rajput, V. Athavale, and S. Mittal, “Intelligent model for classification of SPAM and HAM,” *Int. J. Innov. Technol. Explor. Eng*., vol. 8, no. 6, pp. 773–777, 2019
8. G. Waja, G. Patil, C. Mehta, and S. Patil, “How AI Can be Used for Governance of Messaging Services: A Study on Spam Classification Leveraging Multi-Channel Convolutional Neural Network,” *Int. J. Inf. Manag. Data Insights*, vol. 3, no. 1, p. 100147, 2023, doi:10.1016/j.jjimei.2022.100147
9. Blacklisting vs. Whitelisting. — https://consoltech.com/blog/blacklisting-vs-whitelisting/.
10. REALTIME BLACKHOLE LIST (RBL): WHAT IS IT? https://www.mailinblack.com/en/ressources/blog/rbl-what-isit/.
11. N. N. Amir Sjarif, N. F. Mohd Azmi, S. Chuprat, H. M. Sarkan, Y. Yahya, and S. M. Sam, “SMS spam message detection using term frequency-inverse document frequency and random forest algorithm,” *Procedia Comput. Sci*., vol. 161, pp. 509–515, 2019, doi:

10.1016/j.procs.2019.11.150.
12. Wu J, Deng T. Research in Anti-Spam Method Based on Bayesian Filtering. IEEE, Pacific-Asia Workshop on Computational Intelligence and Industrial Application, pp. 887 – 891, 2008
13. Ozgur L, Gungor T, Gurgen F. Spam Mail Detection Using Artificial Neural Network and Bayesian Filter. Springer, pp. 505-510, 2004.
14. Dong J, Cao H, Liu P, Ren L (2006). Bayesian Chinese Spam Filter Based on Crossed N-gram. Intelligent Systems Design and Applications, 2006. ISDA '06. Sixth International Conference on, 3:103-108.
15. Naive Bayes spam filtering. https://en.wikipedia.org/wiki/Naive_Bayes_ spam_filtering.
16. An Approach to Email Classification Using Bayesian Theorem. https://core.ac.uk/download/pdf/231152837.pdf.
17. Ten Spam-Filtering Methods Explained.: https://www.connectingup.org/learn/articles/ten-spam-filtering-methodsexplained.
18. Spam Filtering Using Bag-of-Words. https://heartbeat.fritz.ai/spam-filtering-using-bag-of-words-1c5484ff07f1.
19. Mohammed MA, Gunasekaran SS, Mostafa SA, Mustafa A, Ghani MKA. Implementing an agent-based multi-natural language anti-spam model. 2018 International Symposium on Agent, Multi-Agent Systems and Robotics (ISAMSR). Putrajaya, Malaysia: IEEE; 2018 Aug. p. 1–5.
20. Akinyelu AA, Adewumi AO. Classification of phishing email using random forest machine learning technique. J Appl Math.2014;2014:425731.
21. Sutta N, Liu Z, Zhang X. A study of machine learning algorithms on email spam classification. *Proceedings of 35th International Conference*. Vol. 69. Missouri, USA: Southeast Missouri State University; 2020. p. 170–9
22. Khamis SA, Foozy CFM, Ab Aziz MF, Rahim N. Header based email spam detection framework using support vector machine (SVM) technique. *International Conference on Soft Computing and Data Mining*. Cham: Springer; 2020 Jan. p. 57–65.
23. S. Chaudhry, S. Dhawan, R. Tanwar, Spam Detection in Social Network Using Machine Learning Approach, in: U. Batra, N. Roy, B. Panda (Eds.), Data Science and Analytics. REDSET 2019, Communications in Computer and Information Science, 2020, pp. 236-245.doi:10.1007/978-981-15-5830-6_20.
24. Ch. Zhao, Y. Xin, X. Li, Y. Yang, Yu. Chen, A Heterogeneous Ensemble Learning Framework for Spam Detection in Social Networks with Imbalanced Data, Applied Sciences, 10, 936, 2020. doi:10.3390/app10030936.
25. N. Sharma, Understanding the Mathematics behind Support Vector Machines, Heartbeat, 2020.URL: https://heartbeat.fritz.ai/understanding-the-mathematics-behind-support-vector-machines-5e20243d64d5.
26. Converting Raw Text to Numerical Vectors using Bag of Words, N Grams and TF-IDF. https://aiaspirant.com/bag-ofwords/.
27. Метод опорних векторів. Режим доступу: https://uk.wikipedia.org/wiki/ Метод_опорних_векторів.

PHISHING DETECTION ALGORITHM IN MESSENGER LETTERS AND EMAILS USING NEURAL NETWORKS WITH A FIXED NUMBER OF RANKING RISK FACTORS

V.O. Nazarov, A.V. Sadchenko, O.A. Kushnirenko

National Odesa Polytechnic University
1, Shevchenko Ave., Odesa, 65044, Ukraine
Emails; koa@op.edu.ua, sadchenko@op.edu.ua

The article is devoted to the development of algorithms for detecting the most harmful type of spam - phishing - based on neural networks with sigmoid and threshold activation functions. A review and analysis of existing approaches to spam detection was conducted and it was found that most modern algorithms use a quantitative approach based on the constant replenishment of dictionaries of word combinations in explicit and tokenized form, which, when using, for example, a neural network at the decision-making stage, can increase the probability of classifying ethical content of messages and letters as spam. A new approach to forming a vector of input factors that can be present in both spam and ethical content is presented. Four main general factors were identified, ranked on a ten-point risk scale: “possible reward”, “time limit”, “emotional coloring” and “presence of a link to an external resource”. Regarding the first 3 factors, a 10-point scale of impact assessment is proposed according to the analysis of the presence of word-combination patterns, and “presence of a link to an external resource” has two gradations: “1” if there is a link, and “0” if there is no link. This ranking allowed us to create a digital representation of risk factors in the form of an input vector of factors, which is then fed to the input of a neural network with sigmoid or threshold activation functions. A neural network using a sigmoid activation function after training, based on ten cases of phishing content and the same number of ethical emails, provided a probability of detecting malicious content of approximately 10^{-3}, provided that suspicious emails are classified as spam. A neural network with a threshold activation function and an additional division of the range of the factor “presence of a link to an external resource” into ten conditional gradations, using the analysis of the S matrix of mutual influence, showed a high learning rate and a probability of detecting malicious content of approximately $0.5 \cdot 10^{-3}$. The results of the research can be used to create phishing filters with both hard and soft solutions, which are intended for integration into email services and social network messengers.

Keywords: spam filters, phishing emails, neural network with sigmoid activation function, risk factors.

DOI 10.15276/imms.v15.no2.260
УДК 004.056.5

Informatics and Mathematical Methods in Simulation
Vol.15 (2025), No. 2, pp. 260-275

КІБЕРІМУННА МОДЕЛЬ ЗАХИСТУ ДАНИХ: МІЖДИСЦИПЛІНАРНИЙ АНАЛІЗ, ДІАГНОСТИКА ТА АРХІТЕКТУРА

О.А. Сиропятов, Л.М. Тимошенко

Національний університет «Одеська політехніка»
1, Шевченка пр., м.Одеса, 65044,Україна
Emails: o.a.syropiatov@op.edu.ua, l.m.timoshenko@op.edu.ua

У статті проведено міждисциплінарний аналіз можливостей застосування принципів біологічної імунної системи та методів медичної діагностики для розробки ефективної архітектури кіберімунного захисту даних у телекомунікаційних системах. Здійснено огляд сучасних методологій, заснованих на принципах біологічного імунітету (Somayaji, Kim & Bentley, Dasgupta & Niño), які, попри здатність до виявлення аномалій, обмежені через відсутність повноцінних протоколів комплексного реагування на загрози. Запропоновано інноваційну адаптацію принципів первинної медичної діагностики — анамнезу, скринінгу, експрес-тестів і клінічних протоколів — для створення багаторівневої системи виявлення, класифікації та локалізації кіберзагроз із подальшою адаптивною відповіддю. Окреслено концептуальну архітектуру кіберімунної моделі, що включає механізми саморозпізнавання, раннього виявлення аномалій, формування кіберпам'яті, багаторівневу адаптивну відповідь та інформаційну зв'язність компонентів системи. Такий підхід дозволяє закласти фундамент для розвитку самонавчальних систем кіберзахисту з підвищеною стійкістю до складних і цілеспрямованих атак. У статті також детально розглянуто обмеження запропонованої моделі та визначено перспективні напрямки подальших досліджень, що включають моделювання, практичну верифікацію та інтеграцію з міжнародними стандартами кібербезпеки.

Ключові слова: кіберімунна система, імунна відповідь, діагностика, кібербезпека, адаптивний захист, архітектура системи, телекомунікаційні мережі, міждисциплінарний підхід.

Вступ. Історія людства демонструє, що прогрес часто є завдяки міждисциплінарному перенесенню знань та досвіду з однієї галузі в іншу. Приклади – від винайдення друкарського верстата, що трансформував освіту, до адаптації біологічних і нейрофізіологічних ідей в обчислювальні технології для розвитку штучного інтелекту [1]. У XXI столітті такі міждисциплінарні підходи, зокрема, застосування еволюційних алгоритмів чи моделей поширення інфекцій для аналізу соціальних процесів, стають ключовим фактором наукового прогресу, дозволяючи вирішувати складні завдання, такі як оптимізація логістики чи кібербезпека.

У сучасних умовах кіберзагрози стають дедалі складнішими, що вимагає нових, більш ефективних підходів до захисту інформаційних систем [2]. Використання принципів медичної діагностики та імунітету для побудови архітектур кіберзахисту є перспективним напрямом, що поєднує досягнення медицини, біології та інформаційних технологій. Такий міждисциплінарний підхід дозволяє створити адаптивні, самонавчальні системи, здатні ефективно протидіяти сучасним кіберзагрозам.

Метою роботи є обґрунтування концепції кіберімунної системи захисту даних у телекомунікаційних системах на основі міждисциплінарного підходу, що поєднує принципи біологічної імунної системи, медичної діагностики та сучасних архітектур кібербезпеки. Для досягнення мети необхідно вирішити наступні завдання.

1. Провести аналітичний огляд і порівняльний аналіз основних підходів до застосування принципів імунної системи у кіберзахисті, визначити їх переваги, обмеження та сучасний стан розвитку.

2. Проаналізувати ключові етапи та методи первинної діагностики в медицині, з'ясувати можливість їх адаптації для раннього виявлення та класифікації кіберзагроз.

3. Розробити концептуальну архітектуру кіберімунної моделі захисту даних у телекомунікаційних системах, визначити її рівні, компоненти, механізми взаємодії та відповідність сучасним стандартам.

4. Систематизувати результати дослідження, сформулювати висновки щодо ефективності міждисциплінарного підходу, окреслити обмеження та перспективи подальших досліджень.

Таким чином, міждисциплінарний підхід, що поєднує імунологічні та медичні принципи, створює нову парадигму для архітектури кіберзахисту.

Аналітичний огляд підходів до застосування імунної системи у кіберзахисті

Використання принципів біологічної імунної системи як концептуальної основи для архітектури кіберзахисту має багаторічну історію розвитку. Цей розділ містить критичний аналіз трьох фундаментальних підходів, які сформували теоретичний базис для штучних імунних систем (AIS) та рішень, заснованих на імунній системі (далі immune-inspired підходи), у сфері безпеки.

Підхід, запропонований Somayaji, Hofmeyr та Forrest, став одним із перших системних спроб формалізувати імунологічні принципи для кібербезпеки [3]. Автори виділили ключові властивості природної імунної системи, які можуть бути адаптовані для комп'ютерних систем, зокрема:

- децентралізація та автономність – відсутність єдиного центру управління, що забезпечує стійкість до атак на окремі компоненти;
- відсутність довіреного ядра – система функціонує без припущення про наявність абсолютно захищених елементів;
- динамічна адаптація – постійне оновлення та навчання детекторів для розпізнавання нових загроз;
- різноманіття механізмів – використання множини методів виявлення для підвищення загальної ефективності.

Somayaji та інші запропонували концептуальну модель, яка передбачає розподілену мережу детекторів, що функціонують автономно та адаптуються до змін середовища. Однак їх модель залишилася на рівні абстрактної архітектури без чіткої формалізації процесів виявлення та реагування. Вони не розробили конкретних алгоритмів для реалізації запропонованих принципів, що ускладнює практичне впровадження їх ідей. Критичні обмеження моделі:

- відсутність формалізованого протоколу реагування з чіткими етапами;
- не визначено механізми координації між компонентами системи;
- не розглянуто питання ресурсоємності та масштабованості;
- відсутня інтеграція з існуючими стандартами управління інцидентами.

Kim та Bentley зосередилися на прикладному аспекті імунологічних принципів, спеціально досліджуючи їх застосування для мережевих систем виявлення вторгнень (IDS) [4]. Їх ключові інновації:

- негативна селекція – формування детекторів, що не реагують на нормальну поведінку системи;
- клональна селекція – розмноження та мутація успішних детекторів для підвищення ефективності;
- самоорганізація агентів – розподілена система легковісних сенсорів без централізованого контролю;
- імунна пам'ять – збереження інформації про попередні атаки для прискорення майбутнього реагування.

Kim і Bentley розробили більш формалізовані алгоритми для генерації та навчання детекторів, що дозволило наблизити теоретичні концепції до практичної реалізації. Їх експерименти показали потенціал для виявлення нових типів аномалій без

попереднього навчання на відомих атаках. Проте їх підхід зосереджувався переважно на фазі виявлення, залишаючи без уваги комплексний процес реагування. Критичні обмеження підходу:

- спрощені моделі імунної відповіді без врахування багаторівневої взаємодії;
- відсутність механізмів контекстуального аналізу подій;
- недостатня увага до процесів ізоляції та нейтралізації загроз;
- проблеми з масштабуванням при збільшенні кількості детекторів.

Dasgupta і Niño систематизували напрям штучних імунних систем як окрему гілку обчислювального інтелекту, створивши теоретичний фундамент для формалізації імунологічних механізмів[5]. Їх ключові досягнення:

- математична формалізація – строгий опис імунних механізмів у вигляді алгоритмів;
- форма-простір – концепція для представлення антигенів і антитіл у багатовимірному просторі;
- еволюційні підходи – використання генетичних алгоритмів для оптимізації детекторів;
- універсальність застосування – розширення сфери використання від безпеки до оптимізації та класифікації.

Dasgupta і Niño створили найбільш формалізований і математично обґрунтований підхід до штучних імунних систем. Їхня робота дозволила перейти від концептуальних аналогій до конкретних обчислювальних моделей з чітко визначеними алгоритмами. Однак, зосереджуючись на алгоритмічній стороні, вони приділили менше уваги архітектурним аспектам та інтеграції з існуючими системами безпеки. Критичні обмеження підходу:

- фокус на окремих алгоритмічних задачах без цілісної архітектури;
- відсутність інтегрованого протоколу реагування з чіткими фазами;
- недостатня увага до практичних аспектів впровадження в реальні системи;
- обмежена підтримка контекстної інформації в системах управління інцидентами.

Між біологічною та кіберімунною системами існують фундаментальні відмінності, які ускладнюють пряму інтеграцію [6]. Структурні відмінності:

- контекст середовища: біологічна система функціонує в межах цілісного організму з чіткими межами, тоді як кіберпростір характеризується розмитими кордонами та динамічною структурою;
- еволюційна історія: природна імунна система розвивалася мільйони років, забезпечуючи оптимальний баланс між ефективністю та ресурсоємністю, тоді як кіберімунні системи проєктуються штучно.
- механізми адаптації: біологічна система має вбудовані механізми самонавчання та адаптації, які складно відтворити в цифровому середовищі.

Функціональні відмінності:

- багаторівневість захисту: природний імунітет включає вроджений та адаптивний компоненти з чітким розподілом функцій, тоді як кіберзахист часто реалізується як набір окремих інструментів;
- формування пам'яті: біологічна система ефективно формує довготривалу імунологічну пам'ять, що складно реалізувати в кіберсистемах без надмірного споживання ресурсів;
- протоколи реагування: медицина має формалізовані протоколи діагностики та лікування, тоді як кіберзахист часто залежить від ad-hoc рішень.

Сучасні виклики та нові парадигми

Впровадження immune-inspired підходів у реальні середовища має сучасні виклики. Технологічні виклики:

- постійно зростаюча швидкість і складність атак;
- необхідність контекстуального аналізу подій у реальному часі;

- інтеграція різнорідних джерел даних (SIEM, IDS/IPS, Threat Intelligence);
- балансування між глибиною аналізу та швидкістю реагування.

Нові парадигми безпеки:

- Zero Trust – модель, що базується на принципі "нікому не довіряй, завжди перевіряй" [7];
- XDR (Extended Detection and Response) – інтегровані платформи для наскрізного виявлення та реагування [8];
- SOAR (Security Orchestration, Automation and Response) – автоматизовані платформи з формалізованими сценаріями реагування[9];
- MITRE ATT&CK – для класифікації та аналізу тактик і технік атак[10].

Ці сучасні рішення, хоча й не використовують біологічні метафори напряму, реалізують принципи адаптивності, автоматизації та координації, необхідні для ефективного кіберзахисту. У таблиці 1 показано переваги та обмеження цих підходів.

Таблиця 1.

Ключові переваги та обмеження підходів

Підхід	Основні ідеї та переваги	Ключові обмеження
Somayaji et al.	Децентралізація, автономність, різноманіття; відсутність єдиної точки відмови; адаптивність до нових загроз	Абстрактність, відсутність формалізованого протоколу реагування; немає інтеграції зі стандартами
Kim & Bentley	Негативна селекція, клонування, самоорганізація; генерація детекторів для IDS; виявлення нових аномалій; простота імплементації	Спрощені моделі, відсутність багатошарової взаємодії; немає комплексного протоколу реагування
Dasgupta & Niño	Формалізація імунних механізмів у вигляді алгоритмів; еволюційні підходи; автоматизація виявлення загроз; адаптивність через навчання	Фокус на окремих задачах; відсутність інтегрованого протоколу реагування; обмежена робота з контекстом
Запропонована кіберімунна модель	Адаптація принципів імунної відповіді та медичної діагностики; багаторівнева архітектура; саморозпізнавання; раннє виявлення аномалій; адаптивна відповідь; формування кіберпам’яті; інформаційна зв’язність компонентів; основа для самонавчальних систем	Потребує подальшого моделювання, практичної верифікації та інтеграції з міжнародними стандартами

Стан застосування та перспективи

Аналіз сучасного стану застосування immune-inspired підходів виявляє суттєві прогалини. Поточні обмеження:

- більшість досліджень зосереджені на алгоритмах аномального виявлення (негативна селекція, клональна селекція, моделі небезпеки);
- AIS використовуються для оптимізації та класифікації, але не забезпечують комплексної взаємодії між компонентами захисту;
- галузеві стандарти (NIST SP 800-61, MITRE ATT&CK, SIEM, SOAR) не інтегрують імунологічні метафори як основу архітектури.

Перспективні напрями розвитку:

− розробка багаторівневих архітектур з чітким розподілом функцій між компонентами;

− формалізація протоколів реагування за аналогією з медичними протоколами лікування;

− інтеграція контекстуального аналізу для підвищення точності виявлення загроз;

− створення механізмів формування "кіберпам'яті" для ефективного навчання на інцидентах.

Проведений аналіз існуючих immune-inspired підходів показує, що, незважаючи на значний теоретичний фундамент, вони є фрагментарними і не забезпечують комплексного вирішення проблем кіберзахисту. Для подальшого розвитку необхідно перейти від окремих алгоритмів до інтегрованих систем управління інцидентами, які поєднують принципи імунної системи на всіх етапах – від виявлення до формування кіберпам'яті. Це вимагає розробки формалізованої архітектури, що адаптує біологічні механізми до специфіки кіберсередовища з урахуванням сучасних викликів і технологічних можливостей.

Аналогії у діях біологічної імунної системи та кіберзахистом

Методи первинної діагностики в медицині

Первинна діагностика є фундаментом сучасної медицини, оскільки дозволяє виявляти захворювання на ранніх стадіях, коли лікування найбільш ефективне, а ризики ускладнень – мінімальні [11]. Вона поєднує в собі як наукові методи, так і мистецтво інтерпретації симптомів у контексті індивідуальних особливостей пацієнта. Сучасна медицина постійно вдосконалює підходи до первинної діагностики, впроваджуючи нові технології та стандарти.

Основні етапи первинної діагностики.

1. Анамнез (збір інформації): передбачає детальний збір відомостей про самопочуття, скарги, історію хвороб, спадковість, спосіб життя пацієнта, формує основу для подальшого аналізу, допомагає окреслити зону пошуку проблеми та визначити пріоритети обстеження.

2. Фізикальне обстеження: безпосереднє вивчення стану організму за допомогою огляду, пальпації, аускультації, перкусії, вимірювання життєвих показників (температура, тиск, пульс). Мета – виявити об'єктивні ознаки порушень та уточнити локалізацію проблеми.

3. Скринінгові дослідження: масові скринінгові тести (наприклад, мамографія, тести на глюкозу, онкомаркери) для виявлення прихованих, безсимптомних або латентних захворювань, дозволяють своєчасно виявити ризики та розробити превентивні заходи.

4. Експрес-тести: у разі необхідності швидкої оцінки стану пацієнта використовуються експрес-методи (швидкі тести на інфекції, біохімічні аналізи). Вони забезпечують оперативне прийняття рішень у невідкладних ситуаціях.

5. Клінічні протоколи: діагностичний процес завершується застосуванням стандартизованих клінічних протоколів – алгоритмів, що регламентують дії лікаря у типових і складних випадках, що забезпечують єдність підходів та підвищують якість медичної допомоги[12].

Важливо, що ефективність первинної діагностики визначається не окремим методом, а їх поєднанням. Синергетичне використання анамнезу, фізикального обстеження, лабораторних тестів і протоколів дозволяє підвищити точність діагнозу та мінімізувати ризик помилок.

Сьогодні до традиційних методів активно додаються цифрові технології – електронні медичні картки, телемедицина, автоматизовані системи підтримки прийняття рішень. Це підвищує швидкість і якість діагностики, а також відкриває нові можливості для інтеграції медичних підходів у суміжні галузі [13].

Системний аналіз методів первинної діагностики в медицині дає змогу визначити

ключові механізми раннього виявлення загроз, що можуть бути адаптовані для підвищення ефективності кібербезпеки.

Імунна відповідь - біологічна основа для кіберімунної моделі.

Побудова кіберімунної моделі неможлива без глибокого розуміння ключових механізмів біологічної імунної системи, які демонструють структурну та функціональну схожість із процесами кіберзахисту [14]. Одним із універсальних та вивчених процесів в імунології є виявлення, обробка та нейтралізація антигену. Саме цей сценарій, що охоплює як вроджений, так і адаптивний імунітет, дозволяє простежити повний цикл захисту – від першого контакту з потенційною загрозою до формування довготривалої пам'яті. У біології антигеном вважають будь-яку молекулу, яка розпізнається імунною системою як "чужа" і здатна викликати імунну відповідь.

Антигени поділяють на екзогенні антигени: віруси, бактерії, токсини, пилок, чужорідні агенти, що потрапляють у організм із зовнішнього середовища; ендогенні антигени: білки змінених клітин, вірусні білки всередині заражених клітин, пухлинні маркери – власні компоненти організму, які зазнали змін або виявляються у незвичних контекстах. Ключовими властивостями антигену є імуногенність – здатність викликати імунну відповідь, та специфічність – забезпечує точність розпізнавання.

На рис.1 показано, як працює імунна система при зустрічі з антигеном.

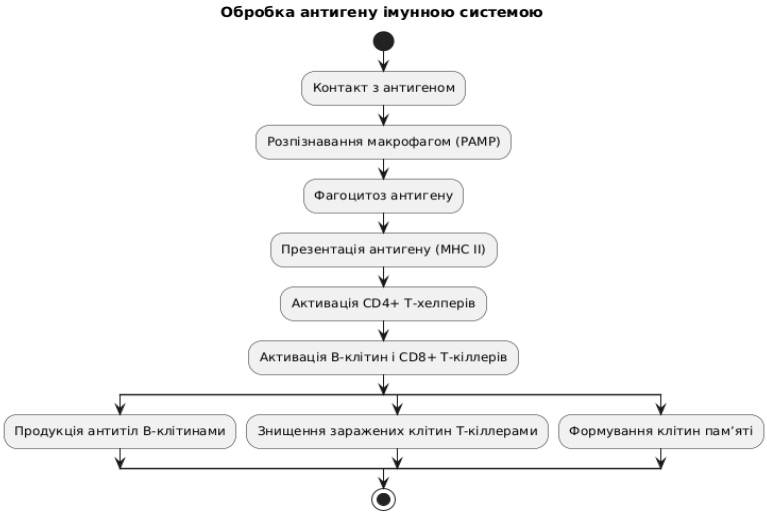


Рис.1. Обробка антигену імунною системою

Етапи імунної відповіді.

1. Розпізнавання антигену (вроджена система): клітини вродженого імунітету (макрофаги, дендритні клітини, нейтрофіли) виявляють характерні патерни (PAMPs), захоплюють антиген (фагоцитоз), переносять його до лімфатичних вузлів і запускають сигнальний каскад.

2. Презентація антигену: оброблений антиген презентується через молекули MHC. MHC класу II – CD4+ Т-хелперам; MHC класу I – CD8+ Т-кілерам.

3. Активація Т-клітин: CD4+ Т-хелпери активують В-клітини (продукують антитіла), макрофаги та інші Т-клітини.

CD8+ Т-кілери знищують клітини-мішені, ініціюючи апоптоз.

4. Вироблення антитіл і клітинна відповідь: антитіла нейтралізують антигени, блокують віруси, позначають їх для фагоцитозу, Т-клітини атакують заражені або атипові клітини.

5. Формування імунної пам'яті: створюються живучі клітини пам'яті (Т- і В-), які забезпечують швидку і потужну вторинну відповідь у разі повторного контакту з тією ж загрозою.

Ключові властивості імунної системи – адаптивность (здатність підлаштовуватися під нові загрози), скоординованость (узгоджена робота різних клітин і механізмів), еволюційноя ефективность (відбір найкращих механізмів захисту).

Здатність імунної системи до точного розпізнавання загроз, запуску цільової реакції та формування стійкої пам'яті є ключовою передумовою для побудови аналогічних захисних механізмів у цифровому середовищі.

Перенесення цих можливостей у цифрову сферу створює передумови для розробки кіберзахисних систем, які не лише реагують на відомі атаки, а й здатні ефективно протистояти новим загрозам завдяки механізмам адаптації та накопичення досвіду[15].

Методи первинного виявлення атак у кібербезпеці.

У сфері кібербезпеки, як і в медицині, ключовим завданням є своєчасне виявлення порушень цілісності складних систем. Хоча об'єкт захисту тут – не фізичне тіло, а інформаційне середовище, принципи виявлення загроз демонструють дивовижну подібність із медичними підходами до діагностики. Інформаційні системи так само піддаються зовнішньому впливу, «інфікуванню», зловживанню та зносу, що вимагає розробки ефективних методів раннього виявлення атак.

Нижче показано основні рівні первинного виявлення атак.

Перший рівень – системне логування («анамнез» інформаційної системи). Системне логування є базовим джерелом інформації про стан цифрового середовища. Воно включає хронологічний запис усіх подій: входів, виходів, помилок, змін конфігурацій. Цей «анамнез» дозволяє аналітикам відстежувати розвиток подій, виявляти аномалії та встановлювати причини інцидентів.

Другий рівень – моніторинг системних показників («життєві параметри»). На цьому рівні здійснюється безперервний моніторинг ключових метрик: навантаження процесора, обсяг переданих даних, кількість підключень, звернень до файлової системи тощо. Подібно до вимірювання температури чи тиску в медицині, ці показники допомагають оцінити, чи перебуває система в стані «інформаційного гомеостазу» або ж виникають відхилення, що сигналізують про потенційну загрозу.

Третій рівень – системи виявлення та запобігання атак (IDS/IPS) («експрес-тести»). Цей рівень передбачає використання IDS/IPS, які швидко реагують на відомі шаблони атак і сигнатури шкідливої поведінки. Хоча їхній діапазон обмежений, ці інструменти дозволяють оперативно виявляти та блокувати найпоширеніші загрози.

Четвертий рівень – сканування вразливостей («скринінгові дослідження»). Подібно до медичних скринінгів, у кібербезпеці застосовуються регулярні сканування вразливостей - спеціальні програми, що перевіряють IT-системи на наявність відомих слабких місць. Це дозволяє виявити потенційні точки входу для зловмисників ще до початку атаки.

П'ятий рівень – автоматизовані сценарії реагування (SIEM/Playbooks) («клінічні протоколи»). Завершальний рівень – автоматизовані сценарії реагування, реалізовані у SIEM-системах. Playbooks визначають послідовність дій у разі певних комбінацій подій: від надсилання сповіщень до автоматичного блокування джерел загроз. Це цифровий аналог клінічних протоколів, що забезпечує стандартизовану та швидку реакцію на інциденти.

Жоден із перерахованих рівнів не є самодостатнім. Ефективність забезпечує лише їх комплексне застосування, коли автоматизовані засоби, логічний аналіз, сканування та поведінковий моніторинг працюють у взаємозв'язку. Такий синергетичний підхід формує надійний фундамент для розвитку адаптивної системи кіберзахисту, здатної до самонавчання й превентивного реагування.

Комплексне застосування різних методів виявлення атак у кібербезпеці забезпечує своєчасне реагування та підвищує стійкість інформаційних систем до сучасних загроз. Взаємодія логування, моніторингу, сканування вразливостей та автоматизованих сценаріїв реагування створює основу для формування адаптивної системи кіберзахисту,

здатної до безперервного удосконалення [16]

Кіберобробка аномалій: структурна аналогія з імунітетом

Аномалія як "антиген" цифрового середовища

У цифровому середовищі аналогом біологічного антигену виступає аномалія – подія або поведінка, що не відповідає стандартному (еталонному) профілю функціонування системи. Виявлення таких аномалій є першим кроком до захисту інформаційної інфраструктури від потенційних загроз.

Типові приклади аномалій – логін у нетиповий час або з незвичної IP-адреси; раптове зростання трафіку без пояснення; запуск невідомого процесу; несанкціонована зміна конфігурації.

На рис.2 показано, як працює кіберзахист при виявленні та реагуванні на атаку.

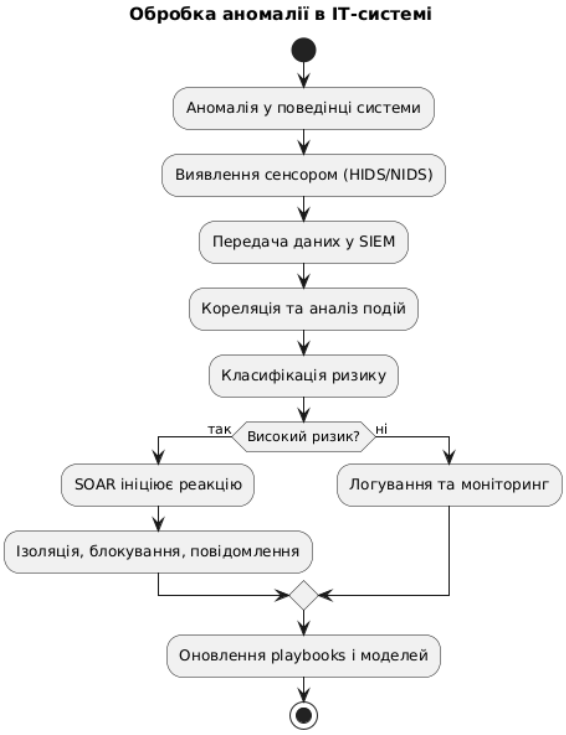


Рис.2. Виявлення та реагування кіберзахисту на зовнішню атаку

Етапи виявлення та реагування, кібернетична імунна відповідь:

- спостереження і моніторинг: агенти моніторингу (HIDS/NIDS, логери, сенсори) безперервно сканують середовище в реальному часі, збираючи дані з мережі (пакети, порти), хостів (процеси, логи), а також із зовнішніх джерел (SOC feed);
- виявлення і кореляція: аналітичні платформи (SIEM, XDR) корелюють події, зіставляють їх із шаблонами та використовують машинне навчання для виявлення нетипових дій, що можуть свідчити про атаку;
- оцінка та класифікація: визначається характер події (шкідлива/нешкідлива), рівень ризику та відповідність сценаріям реагування; у складних випадках подія передається на ручну оцінку SOC-аналітику.
- реакція: активується SOAR або вручну ініціюються дії: блокування IP-адреси, ізоляція пристрою, оновлення фаєрволу, запуск скриптів реагування;
- формування кіберімунної пам’яті: оновлюються правила, сигнатури, ML-моделі, сценарії playbooks, система накопичує досвід для швидшого і точнішого реагування на аналогічні загрози в майбутньому.

Нижче у таблиці 2 наведено порівняння ключових компонентів біологічної імунної системи та відповідних механізмів кібербезпеки, що ілюструє аналогії в процесах виявлення та реагування на загрози.

Таблиця 2.

Аналогія між біологічною імунною системою та кіберзахистом

Імунологія	Кібербезпека	Функція
Антиген	Аномалія	Запуск реакції
Макрофаг	Сенсор/агент моніторингу	Первинне виявлення
Презентація (MHC)	Кореляція в SIEM	Передача у систему аналізу
Т-клітини	SOAR/аналітик	Класифікація та відповідь
Пам’ять	Playbooks, сигнатури	Повторне реагування

Аналіз механізмів виявлення та нейтралізації загроз у біологічних і цифрових системах підтвердив глибоку структурну і функціональну схожість між імунною відповіддю організму та алгоритмами кіберзахисту. Як у медицині, так і в кібербезпеці, ефективний захист базується на поетапному виявленні порушень, їх точному розпізнаванні, адекватній реакції та збереженні досвіду для подальшого самонавчання.

Ці паралелі створюють концептуальне підґрунтя для розробки кіберімунної архітектури, яка здатна самостійно виявляти аномалії; класифікувати джерела загроз; обирати рівень і форму реагування; накопичувати та використовувати кіберпам’ять.

Таким чином, узагальнення виявлених відповідностей дозволяє перейти від описових аналогій до побудови цілісної ідеології кіберімунної системи, що є основою для подальшої практичної реалізації цієї моделі у цифровому середовищі [17].

Архітектура кіберімунної моделі захисту.

Попередній аналіз виявив суттєві концептуальні паралелі між імунною відповіддю біологічних систем та методами кіберзахисту. Ці паралелі слугують підґрунтям для формалізації цілісної кіберімунної моделі – інноваційної архітектури, що інтегрує біологічні принципи в інформаційну безпеку [18].

Перейдемо до розробки концептуальної архітектури кіберімунної моделі, здатної інтегруватися в існуючі системи кіберзахисту з метою підвищення їх ефективності, адаптивності та стійкості до нових типів загроз.

Архітектура кіберімунної моделі базується на основних принципах, адаптованих із біологічної імунної системи, що забезпечують її ефективність, гнучкість і стійкість.

1. Саморозпізнавання і самовизначення (Self-awareness): система повинна чітко визначати власний «нормальний стан» для своєчасного виявлення відхилень і потенційних загроз.
2. Раннє виявлення аномалій (Early Detection): постійний моніторинг, евристичний аналіз та використання методів машинного навчання і штучного інтелекту для швидкого виявлення нетипових подій.
3. Адаптивність (Adaptability): самонавчання та динамічне оновлення алгоритмів реагування відповідно до появи нових загроз і змін у середовищі.
4. Мінімізація хибнопозитивних результатів: валідація дій на основі контекстного аналізу та крос-даних для зниження кількості помилкових спрацювань і підвищення точності.
5. Багаторівнева відповідь (Layered Response): диференційовані реакції залежно від рівня ризику, типу загрози та контексту подій.
6. Формування кіберпам’яті (Cyber Memory): накопичення досвіду, збереження ефективних шаблонів реагування та їх використання для підвищення ефективності майбутніх дій.
7. Інформаційна зв’язність (Information Integration): забезпечення обміну та узгодженого аналізу даних між усіма компонентами системи для цілісного розуміння ситуації.
8. Стійкість до внутрішніх збоїв (Fault Tolerance): здатність системи зберігати функціональність навіть при частковій відмові окремих компонентів.

Ці принципи забезпечать надійність, гнучкісь та масштабованість системи кіберзахисту та відповідають сучасним підходам до побудови адаптивних систем виявлення та реагування.

Для практичної реалізації кіберімунної моделі важливо чітко продемонструвати, як саме принципи втілюються у цифровому середовищі. Таблиця 3 систематизує відповідність між біологічними принципами імунітету та їх реалізацією в архітектурі кіберзахисту, ілюструючи алгоритм роботи моделі на кожному етапі. Тут кожен принцип імунітету має свою функціональну реалізацію в архітектурі кіберімунної моделі. Запропонований алгоритм передбачає послідовне проходження етапів – від самовизначення до формування пам'яті та адаптації, що дозволяє системі не лише реагувати на відомі загрози, а й навчатися на нових інцидентах [19].

Таблиця 3.

Алгоритм імунної відповіді в кіберпросторі: реалізація принципів

Принцип біологічного імунітету	**Реалізація в кіберпросторі**	**Опис/Функція**
Саморозпізнавання і самовизначення	Визначення еталонного стану системи	Формування профілю «нормальної» поведінки, базових політик безпеки, білих списків тощо
Раннє виявлення аномалій	Постійний моніторинг, евристика, ML/AI	Виявлення відхилень у реальному часі, запуск аналізу при появі підозрілих подій
Адаптивність	Динамічне оновлення правил, самонавчання	Впровадження алгоритмів, що змінюють поведінку системи відповідно до нових загроз
Мінімізація хибнопозитивних результатів	Контекстний аналіз, крос-верифікація	Перевірка підозрілих подій за декількома джерелами, зниження кількості помилкових спрацювань
Багаторівнева відповідь	Диференційовані сценарії реагування	Реакція залежно від рівня ризику: від сповіщення до ізоляції чи блокування
Формування кіберпам'яті	Оновлення бази сигнатур, шаблонів, playbooks	Збереження даних про атаки, автоматичне оновлення механізмів реагування
Інформаційна зв'язність	Інтеграція даних між компонентами	Обмін інформацією між сенсорами, аналітичними модулями, системами реагування
Стійкість до внутрішніх збоїв	Резервування, децентралізація	Забезпечення роботи системи навіть за відмови окремих вузлів або каналів

Такий підхід базується на багаторівневій організації, адаптивності та здатності до накопичення кіберпам'яті. Комплексна реалізація підходу забезпечує гнучкість, стійкість і самонавчання системи у динамічному кіберсередовищі .

Завдяки структурі кіберімунна система здатна не лише імітувати біологічні механізми захисту, а й підвищувати ефективність реагування на нові, ще невідомі типи атак, що створює потужне підґрунтя для подальшої деталізації структурних компонентів моделі, які розглядаються далі.

Структура кіберімунної моделі базується на принципах багаторівневої організації та розподілу функцій, що детально описані у фундаментальних роботах з штучних імунних систем (AIS) [5]. Для реалізації цих принципів кіберімунна система має містити такі основні компоненти з відповідними біологічними аналогіями:

− сенсори моніторингу (аналог макрофагів) які постійно сканують цифрове середовище, виявляють аномалії та передають інформацію для подальшого аналізу;

− аналітичний центр (аналог лімфатичних вузлів) здійснює кореляцію подій, класифікацію загроз і приймає рішення щодо необхідних заходів реагування;

− виконавчі модулі реагування (аналог Т-клітин та антитіл) виконують блокування, ізоляцію, відновлення або інші дії залежно від типу та рівня загрози;

− блок кіберпам'яті (аналог клітин пам'яті) зберігає шаблони атак, сценарії реагування та моделі машинного навчання для швидкого і ефективного повторного реагування;

− інтеграційний модуль забезпечує обмін даними між компонентами системи, а також інтеграцію з іншими системами кіберзахисту (SIEM, SOAR, XDR тощо).

Ця багаторівнева модульна архітектура дозволяє забезпечити цілісність, гнучкість та стійкість системи кіберзахисту, а також адаптивність до нових типів загроз у динамічному інформаційному середовищі [20].

Архітектура кіберімунної моделі розробляється з урахуванням вимог міжнародних стандартів інформаційної безпеки: ISO/IEC 27001, NIST SP 800-61 та ISO 15189. Це забезпечує відповідність системи найкращим практикам управління безпекою, моніторингу, виявлення інцидентів та реагування на них, сприяє уніфікації процедур кіберзахисту, полегшує сертифікацію системи та підвищує довіру користувачів і партнерів.

Для коректної адаптації біологічних механізмів у цифровому середовищі сформовано глосарій відповідностей між ключовими елементами імунної системи та їх кібернетичними аналогами. Таблиця 4 демонструє відповідність між біологічними механізмами та кіберімунними компонентами, що дозволяє будувати ефективні системи виявлення та реагування на аномалії.

Таблиця 4.

Відповідність між біологічними механізмами та кіберімунними компонентами

Біологічний аналог	**Кіберімунний компонент**	**Функціональна роль**	**Приклад систем**	**Коротке пояснення**
Макрофаги	Агенти моніторингу	Первинне виявлення загроз	HIDS, SNMP-агенти	Виявляють аномалії на рівні вузлів і мережі
Дендритні клітини	Комутатори, логери	Збір та передача подій	NetFlow-колектори, лог-сервери	Агрегують і пересилають дані для подальшого аналізу
Т-хелпери (CD4+)	AI/SOC класифікатор	Вибір типу реагування	SIEM з ML, SOC-аналітик	Класифікують загрози, визначають сценарії реагування
Т-кілери (CD8+)	Реактивні механізми	Ізоляція, блокування, знищення	SOAR, автоматичне блокування	Здійснюють активну відповідь: блокують

Біологічний аналог	Кіберімунний компонент	Функціональна роль	Приклад систем	Коротке пояснення
				трафік, ізолюють вузли
В-клітини	Генератор сигнатур/правил	Створення шаблонів для виявлення	IDS/IPS, генерація YARA-правил	Формують нові правила для розпізнавання атак
Клітини пам'яті	Playbooks, архів інцидентів	Повторне реагування	Бази інцидентів SOC, бібліотеки playbooks	Зберігають досвід для швидкої реакції на повторні загрози
Кістковий мозок	База знань, система оновлень	Підживлення моделі новими даними	Централізовані репозиторії IOC	Постачають нові знання про загрози та способи реагування
Кровоносна система	Канали телеметрії	Обмін інформацією між модулями	Syslog, MQTT, REST API	Забезпечують циркуляцію даних у системі
Нейронна система	Центр управління	Координація, прийняття рішень	NOC/SOC, оркестратори мережі	Координують процеси та приймають стратегічні рішення

Таблиця ілюструє структурну та функціональну відповідність між ключовими елементами біологічної імунної системи та сучасними компонентами кіберзахисту у телекомунікаційних мережах. Використання реальних прикладів (HIDS, SOAR, SIEM, playbooks, репозиторії IOC, NOC/SOC) демонструє, як міждисциплінарна аналогія переходить у практичну площину. Ця відповідність є основою для побудови ефективної, самонавчальної та адаптивної кіберімунної системи, яка не лише імітує біологічні механізми, а й підсилює можливості сучасних телекомунікаційних платформ у протидії складним загрозам.

Візуалізація архітектури кіберімунної моделі.

Для кращого розуміння логіки взаємодії структурних компонентів кіберімунної моделі доцільно представити її архітектуру у вигляді структурної схеми (рис. 3). На схемі відображено основні функціональні блоки системи, їхні біологічні аналоги та інформаційні зв'язки, що забезпечують цілісність і адаптивність моделі. Схема ілюструє взаємодію основних компонентів: сенсорів моніторингу, аналітичного центру, виконавчих модулів реагування, системи кіберпам'яті, бази знань, каналів телеметрії та центру управління. Кожен елемент виконує функцію, аналогічну до відповідної структури імунної системи організму, а їхня взаємодія забезпечує самонавчання, адаптацію та стійкість до динамічних кіберзагроз.

Представлена архітектура кіберімунної моделі захисту даних загалом відповідає ключовим біологічним принципам імунної системи, адаптованим для цифрового середовища.

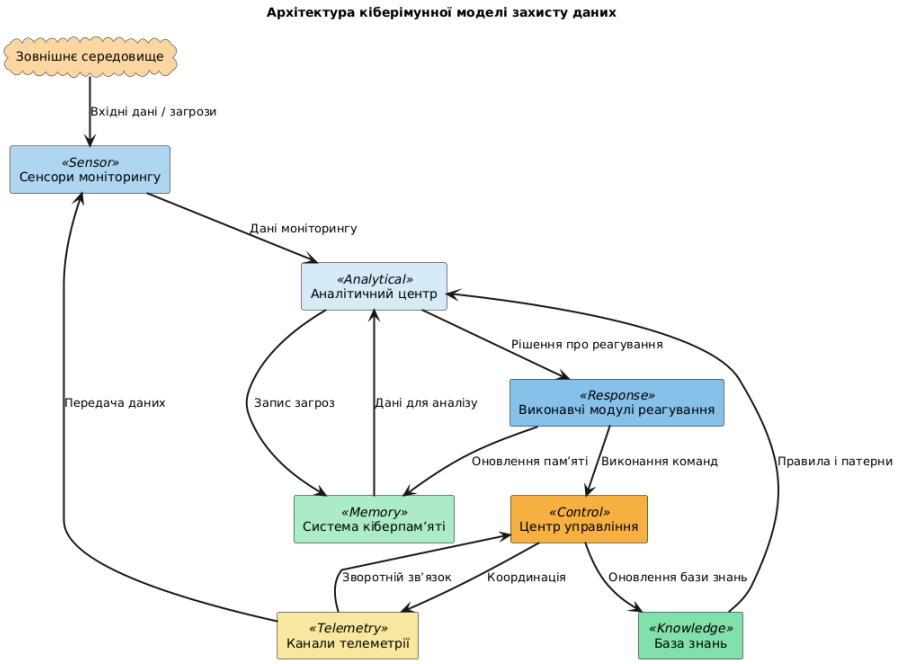


Рис. 3. Архітектура кіберімунної моделі захисту даних у комунікаційних системах

Розглянемо відповідність по кожному з основних принципів.

1. Саморозпізнавання і самовизначення (Self-awareness): архітектура включає сенсори моніторингу, які збирають дані про «нормальний стан» системи, а аналітичний центр формує профіль нормальної поведінки. База знань і кіберпам'ять підтримують збереження еталонних станів і політик.
2. Раннє виявлення аномалій (Early Detection) : постійний моніторинг сенсорами та аналітичний центр із застосуванням евристик, машинного навчання та штучного інтелекту забезпечують швидке виявлення відхилень і підозрілих подій.
3. Адаптивність (Adaptability): кіберпам'ять і база знань оновлюються на основі нових інцидентів і шаблонів реагування, а динамічне оновлення правил у аналітичному центрі й центрі управління забезпечує самонавчання і адаптацію системи.
4. Мінімізація хибнопозитивних результатів: архітектура передбачає контекстний аналіз і крос-верифікацію даних між компонентами (аналітичний центр, база знань, кіберпам'ять), що дозволяє знижувати кількість помилкових спрацювань.
5. Багаторівнева відповідь (Layered Response): диференційні реакції реалізуються через послідовність компонентів: аналітика формує рішення, центр управління координує дії, виконавчі модулі реагування реалізують різні сценарії залежно від рівня загрози.
6. Формування кіберпам'яті (Cyber Memory): система кіберпам'яті зберігає інформацію про атаки та ефективні шаблони реагування, що використовується для покращення майбутніх реакцій.
7. Інформаційна зв'язність (Information Integration): архітектура забезпечує обмін даними між усіма компонентами через канали телеметрії та внутрішні зв'язки, що гарантує цілісний аналіз і координацію дій.
8. Стійкість до внутрішніх збоїв (Fault Tolerance): для реалізації цього принципу доцільно впровадити такі технічні механізми:

- кластеризація ключових компонентів (аналітичний центр, система кіберпам'яті, база знань) із розподіленим навантаженням, що запобігає втраті працездатності при виході з ладу окремих вузлів.
- реплікація даних у реальному часі з георепліkацією або через системи високої доступності (HA), що забезпечує збереження критичних даних навіть при втраті

доступу до окремих дата-центрів.

− Failover-сценарії та автоматичне перемикання на резервні ресурси з мінімальним часом відновлення (RTO) і допустимим вікном втрати даних (RPO).

− самодіагностика та автоматичне відновлення: регулярна перевірка стану компонентів, виявлення деградації продуктивності, ініціювання перезапуску або перенесення процесів на резервні вузли.

− логічна децентралізація функцій для уникнення єдиної точки відмови (SPOF), з дублюванням логіки прийняття рішень у кількох сегментах системи (наприклад часткове дублювання аналітичного центру в регіональних вузлах).

Архітектура відповідає основним принципам кіберімунної моделі, демонструючи послідовний алгоритм імунної відповіді: від самовизначення нормального стану, раннього виявлення аномалій, через адаптивний аналіз і багаторівневу реакцію, до формування кіберпам'яті та забезпечення інформаційної зв'язності. Вона відображає цифрову реалізацію біологічних механізмів імунітету, що забезпечує гнучкість, самонавчання і ефективний захист у динамічному кіберсередовищі.

Висновки. У роботі проведено комплексний міждисциплінарний аналіз теоретичних, методологічних та архітектурних основ побудови кіберімунної моделі захисту даних.

1. Теоретико-методологічний внесок: проведено системний огляд ключових immune-inspired підходів до кіберзахисту. Виявлено, що більшість існуючих рішень фрагментарно застосовують біологічні метафори, зосереджуючись переважно на виявленні аномалій, не охоплюючи повний цикл реагування та навчання.

2. Міждисциплінарний підхід: продемонстровано ефективність адаптації принципів первинної медичної діагностики для виявлення та класифікації кіберзагроз, що підвищує точність і швидкість реагування.

3. Архітектурні рішення: запропоновано модульну архітектуру кіберімунної моделі, що базується на ключових біологічних принципах імунної системи, адаптованих для цифрового середовища. Архітектура забезпечує децентралізацію, автономність, адаптивність, а також високу стійкість завдяки механізмам кластеризації, реплікації даних і автоматичного відновлення. Це сприяє масштабованості системи та підвищує її ефективність у протидії складним багатовекторним атакам.

4. Відповідність міжнародним стандартам: розробка архітектури здійснюється з урахуванням вимог міжнародних стандартів інформаційної безпеки, зокрема NIST SP 800-61 та ISO/IEC 27001, які визначають найкращі практики управління інцидентами, моніторингу, реагування та підтримки безперервності бізнес-процесів. Це забезпечує уніфікацію процедур кіберзахисту, полегшує сертифікацію та впровадження системи.

5. Практичне значення: обґрунтовано доцільність впровадження кіберімунних підходів до підвищення стійкості інформаційних систем до багатовекторних загроз.

6. Обмеження та перспективи: визначено виклики, пов'язані зі складністю формалізації біологічних процесів, необхідністю адаптації до динамічних умов і ризиками хибних спрацювань. Подальші дослідження планується зосередити на моделюванні, симуляціях і практичній верифікації [21].

7. Наукова та практична новизна: результати створюють основу для розвитку нового класу адаптивних, самонавчальних систем кіберзахисту, здатних автономно виявляти, ідентифікувати та нейтралізувати загрози у реальному часі.

8. Перспективи подальших досліджень: це перша частина комплексного дослідження, присвяченого розробці кіберімунної моделі захисту даних. У ній зосереджено увагу на теоретичних і методологічних засадах, а також на архітектурних рішеннях. Далі планується представити результати валідації сформульованих теоретичних положень, а також запропонувати практичні напрямки вдосконалення методів виявлення загроз із застосуванням розробленої кіберімунної моделі. Це дозволить оцінити ефективність підходу в реальних умовах та сприятиме подальшому розвитку адаптивних і самонавчальних систем захисту.

Список літератури

1. Smith J. *History of Interdisciplinary Science*. Берлін: Springer, 2018. 320 с.
2. European Union Agency for Cybersecurity (ENISA). *Threat Landscape 2024*. Athens: ENISA, 2024. 150 с.
3. Somayaji A., Hofmeyr S.A., Forrest S. *Principles of a Computer Immune System*. Proceedings of the National Information Systems Security Conference (NISSC). Baltimore, USA, 1997. С. 335–345.
4. Kim J., Bentley P.J. *Towards an Artificial Immune System for Network Intrusion Detection: Anomaly Detection Using Dendritic Cells*. Proceedings of the Genetic and Evolutionary Computation Conference (GECCO). San Francisco, USA, 2001. С. 12–19.
5. Dasgupta D., Niño L.F. *Artificial Immune Systems and Their Applications*. New York: Springer, 2009. 264 с.
6. Stepney S., et al. *Concepts of Biological Immunity and Computer Security*. Theoretical Computer Science. Amsterdam: Elsevier, 2005.– Т. 337, № 1–3. С. 131–156.
7. National Institute of Standards and Technology (NIST). *Special Publication 800-207. Zero Trust Architecture*. Gaithersburg, USA: NIST, 2020. 59 с.
8. Gartner. *Innovation Insight for Extended Detection and Response (XDR)*. Stamford: Gartner Research, 2022. 35 с.
9. IBM Security. *SOAR: Security Orchestration, Automation and Response*. Armonk, USA: IBM White Paper, 2023. 28 с.
10. MITRE. *ATT&CK Framework*. Режим доступу: https://attack.mitre.org/ (дата звернення: 03.06.2025).
11. World Health Organization (WHO). *Guidelines on Diagnostic Procedures*. Geneva: WHO Press, 2023. 88 с.
12. Centers for Disease Control and Prevention (CDC). *Guidelines for Diagnostic Tests*. Atlanta, USA: CDC, 2022. 76 с.
13. ISO 15189:2022. *Medical laboratories – Requirements for quality and competence*. Geneva: ISO, 2022. 46 с.
14. Timmis J., Neal M., Hunt J. *An Artificial Immune System for Data Analysis*. Biosystems. – Amsterdam: Elsevier, 2000. Т. 55, № 1–3. С. 143–150.
15. Johnson M. *Interdisciplinary Approaches in IT*. IT Review. London: IT Press, 2022. Т. 12, № 4. С. 45–58.
16. ISO/IEC 27001:2022. *Information Security Management Systems – Requirements*. Geneva: ISO, 2022. 35 с.
17. Hart E., Timmis J., Hone A. *Artificial Immune Systems*. In: Gendreau M., Potvin J.-Y. (eds.) *Handbook of Metaheuristics*. – 3rd ed. New York: Springer, 2019. С. 421–448.
18. Hofmeyr S.A., Forrest S. *Architecture for an Artificial Immune System*. Evolutionary Computation. Cambridge, USA: MIT Press, 2000. – Т. 8, № 4. С. 443–473.
19. ISO/IEC 27035-1:2016. *Information Security Incident Management Part 1: Principles of Incident Management*. Geneva: ISO/IEC, 2016. 28 с.
20. ETSI GS NFV 002 V1.2.1. *Network Functions Virtualisation (NFV); Architectural Framework*. Sophia Antipolis: ETSI, 2014. 72 с.
21. Berner E.S., La Lande T.J. *Clinical Decision Support Systems: Successes and Failures*. Yearbook of Medical Informatics. Schattauer, 2016. Т. 25, № 1. С. 103–110.

A CYBER IMMUNE MODEL OF DATA PROTECTION: INTERDISCIPLINARY ANALYSIS, DIAGNOSTICS, AND ARCHITECTURE

O.A. Syropiatov, L.M. Tymoshenko

National Odesa Polytechnic University
1, Shevchenko Ave., Odesa, 65044, Ukraine
Emails: o.a.syropiatov@op.edu.ua, l.m.timoshenko@op.edu.ua

The article presents an interdisciplinary analysis of the possibilities of applying the principles of the biological immune system and medical diagnostic methods to develop an effective architecture for cyberimmune data protection in telecommunications systems. A review of modern methodologies based on the principles of biological immunity (Somayaji, Kim & Bentley, Dasgupta & Niño) is carried out, which, despite the ability to detect anomalies, are limited due to the lack of full-fledged protocols for a comprehensive response to threats. The article proposes an innovative adaptation of the principles of primary medical diagnosis - history, screening, rapid tests and clinical protocols - to create a multi-level system for detecting, classifying and localizing cyber threats with a subsequent adaptive response. The author outlines the conceptual architecture of the cyber immune model, which includes mechanisms for self-recognition, early detection of anomalies, formation of cyber memory, multi-level adaptive response, and information connectivity of the system components. This approach allows laying the foundation for the development of self-learning cyber defense systems with increased resistance to complex and targeted attacks. The article also discusses in detail the limitations of the proposed model and identifies promising areas for further research, including modeling, practical verification, and integration with international cybersecurity standards.

Keywords: cyber-immune system, immune response, diagnostics, cybersecurity, adaptive defense, system architecture, telecommunication networks, interdisciplinary approach.

DOI 10.15276/imms.v15.no2.276
УДК 004.89

Informatics and Mathematical Methods in Simulation
Vol.15 (2025), No. 2, pp. 276-287

ІНСТРУМЕНТИ ШТУЧНОГО ІНТЕЛЕКТУ В РОЗРОБЦІ ГРАФІЧНИХ ЕЛЕМЕНТІВ, 3D-МОДЕЛЕЙ ТА ІНШИХ КОМПОНЕНТІВ ВІДЕОІГОР

О.Г. Трофименко, О.В. Задерейко, Н.І. Логінова, О.В. Шевченко

Національний університет "Одеська юридична академія"
23, Фонтанська дорога, м. Одеса, 65009, Україна
Email: trofymenko@onua.edu.ua

Статтю присвячено дослідженню практичних аспектів використання різних інструментів на базі штучного інтелекту (ШІ) в розробці 3D-моделей та відеоігор. Метою статті є дослідження й систематизація прикладних аспектів застосування ШІ у розробці відеоігор та ігровому процесі, оскільки наявні інструменти ШІ мають різні сфери застосування та різні підходи до генерації контенту, тестування й оптимізації. Методика дослідження полягає в розгляді можливих варіантів використання тих чи інших сучасних інструментів ШІ на різних етапах розробки 3D-гри. В роботи проаналізовано вплив інструментів ШІ на всі етапи та аспекти розробки ігор на прикладі створення гри «Labyrinth of Darkness». Використання інструментів на базі ШІ, таких як Meshy.ai, Leonardo.Ai, AIVA, DeepMotion, GitHub Copilot, Replica Studios та іншим, дозволило знизити трудомісткість розробки, скоротити часові витрати і, що не менш важливо, покращити кінцевий результат. У сукупності ці інструменти дозволили ефективно реалізувати ігровий проєкт і створити цілісну, атмосферну гру з унікальним стилем і захоплюючим ігровим процесом. Особлива увага приділена дослідженню можливостей оптимізації процесу тестування та перевірки програмного коду за допомогою ШІ. Застосування ШІ стало ключовим елементом у процесі створення гри «Labyrinth of Darkness», значно прискоривши розробку та підвищивши якість кінцевого продукту. Сучасні технології ШІ радикально трансформують всі етапи розробки відеоігор, відкривши нові можливості для оптимізації робочих процесів та підвищення якості кінцевого продукту. Завдяки впровадженню цих інноваційних рішень розробники отримали можливість зосередитись на творчості, а рутинні операції генерації 3D-моделей, створення анімацій, написання коду та локалізації текстів стали більш автоматизованими та точними. Впровадження інструментів ШІ в розробку ігор не лише підвищує ефективність виробничих процесів, а й допомагає вирішувати соціальні проблеми галузі, зокрема надмірне навантаження на співробітників («crunch»).

Ключові слова: штучний інтелект, ШІ, інструменти ШІ, GameDev, розробка ігор, тестування, графічний елемент, 3D-моделювання, 3D-модель, графіка, текстура, дизайн, анімація.

Вступ. Наразі не йдеться про те, чи буде штучний інтелект (ШІ) використовуватися в індустрії GameDev, чи ні, оскільки він вже активно використовується як в розробці, так і в тестуванні 3D-ігор. Перспективи використання ШІ в GameDev надзвичайно широкі, і вони вже активно змінюють індустрію. На ринку з’явилися ШІ-інструменти, які допомагають автоматизувати процеси, покращити якість контенту та оптимізувати геймплей. Використання цих інструментів дає розробникам можливість швидше створювати контент, покращувати поведінку некерованих гравцем персонажів (Non-Player Character, NPC) у комп'ютерних іграх, автоматизувати тестування та оптимізувати процеси розробки. Це не лише економить час, а й дозволяє створювати більш складні та реалістичні ігрові світи.

Дослідження прикладних аспектів використання ШІ в розробці ігор є вельми актуальним, оскільки інструменти ШІ здатні суттєво змінити ігрову індустрію та відкривають нові можливості для розробників.

Аналіз досліджень і публікацій. Стаття [1] аналізує переваги й недоліки технології ШІІ, а також розглядає технічні та етичні проблеми, що можуть виникнути в процесі її

впровадження в розробку відеоігор. Дослідження [2] оцінює потенціал використання GPT (Generative Pre-trained Transformer) у відеоіграх. Для цього автори провели огляд 131 публікації, 76 з яких вийшли 2024 року, і визначили 5 основних напрямів застосування GPT: процедурне генерування контенту, дизайн ігрових механік зі змішаною ініціативою, інтерактивний геймплей, використання ШІ для гри та аналіз поведінки гравців. Автор роботи [3] проаналізував вплив ШІ на вдосконалення реакції неігрових персонажів (NPC) у відеоіграх. У статті [4] досліджено різні аспекти застосування великих мовних моделей (Large Language Model, LLM) у геймдеві. У роботі [5] розглянуто використання LLM для генерації рівнів гри Sokoban та залежність ефективності таких моделей від обсягів навчальних даних. У дослідженні [6] проаналізовано, як LLM та GPT можуть виступати в ролі високорівневих творчих асистентів у процесі геймдизайну. Автори статті [7] дослідили можливості LLM для створення прототипів карткових ігор і розробили систему, яка забезпечує узгоджену генерацію ігрового коду, прискорюючи процес прототипування та зменшуючи навантаження на розробників. У дослідженні [8] систематизовано алгоритми ШІ за їх потенційним застосуванням у створенні відеоігор та організації ігрового процесу. У статті [9] презентовано інструмент генеративного ШІ (GenAI), який допомагає дизайнерам у створенні прототипів ігрових персонажів. У роботі [10] представлено систему розпізнавання емоцій на основі нечіткої логіки та нейронних мереж, яка може застосовуватись для динамічного процедурного генерування контенту з механікою інтерактивної взаємодії. Дослідження [11] розглядає можливості та виклики, пов'язані з інтеграцією ШІ у процес дизайну та розробки відеоігор.

Численні публікації, спрямовані на висвітлення проблемних аспектів та можливостей використання ШІ для розробки ігор, свідчить про високий рівень зацікавленості та актуальність дослідження можливих сфер застосування ШІ в індустрії GameDev. При цьому у досліджених публікаціях не приділяється увага прикладним аспектам використання ШІ-інструментів в GameDev.

Метою статті є дослідження та систематизація прикладних аспектів застосування ШІ у розробці відеоігор та ігровому процесі, оскільки наявні інструменти ШІ мають різні сфери застосування та різні підходи до генерації контенту, тестування й оптимізації.

Створення 3D-ігор – це поєднання технічної майстерності, художнього бачення та глибокого розуміння геймплею. Процес розробки потребує синергії між програмістами, художниками, аніматорами, геймдизайнерами та тестувальниками, адже кожен аспект гри має бути не просто функціональним, а й захопливим для гравця.

ШІ відіграє важливу роль на різних етапах розробки 3D-ігор, від створення відео- та аудіоконтенту, 3D-моделей та сценаріїв до генерації програмного коду й тестування (рис. 1).

Рис. 1. Сучасні інструменти на базі ШІ, що використовуються у GameDev

Згідно з опитуванням [12], проведеним на початку 2025 року, 45% розробників ігор вже використовують інструменти генеративного ШІ для моделювання ігрового процесу, що відзначилось зростанням продуктивності роботи на понад 20%. Це свідчить про те, що цей підхід має потенціал значно змінити індустрію GameDev, зробивши розробку швидшою, ефективнішою та доступнішою.

1. Інструменти для генерації 3D-моделей та текстур.

Графічний інтерфейс користувача є ключовим елементом будь-якого програмного продукту, а для відеоігор він формує перше враження про продукт та забезпечує взаємодію користувача із системою. Сучасні ШІ-інструменти трансформують підхід до розробки UI/UX за рахунок швидкого створення якісних візуальних компонентів без потреби залучення великої команди дизайнерів. Це особливо цінно для невеликих команд та інді-розробників, які прагнуть оперативно випустити продукт, оптимізувати витрати на створення інтерфейсу та зосередитись на функціональній частині проєкту. Наприклад, платформа Leonardo.Ai (https://leonardo.ai/) надає широкий набір ШІ-інструментів для створення узгоджених стилістичних елементів, а також можливість створювати ілюстрації та концепти персонажів на основі текстових описів або ескізів розробника.

Процес створення 3D-моделей для гри «Labyrinth of Darkness» починався з пошуку ідей та референсних матеріалів з метою визначення загальної концепції візуальних образів персонажів. Так, для моделі «скелетний лицар» були зібрані референсні матеріали, скетчі, фотографії та концепт-арти. Його концептуальне зображення було згенероване засобами Leonardo.Ai, де за допомогою ключових слів з описом бажаного вигляду, а саме: «скелетний лицар», «темна броня», «готичний стиль», отримали зображення (рис. 2), яке стало візуальною основою для подальшого моделювання.

Рис. 2. Концептуальне зображення скелету лицаря, створене за допомогою Leonardo.Ai

Для генерації окремих візуальних елементів ігрового середовища (зомбі-ворогів, зброї, пасток, факелів, вогнищ тощо) у проєкті «Labyrinth of Darkness» використано інструмент Leonardo.Ai із застосуванням шаблона «Game Concept» з високою контрастністю, що забезпечило чіткість і насиченість зображень. Було згенеровано по декілька варіантів візуальних елементів, щоб розробники вибрали найбільш оптимальні варіанти для подальшої адаптації та інтеграції у візуальне оформлення гри. Вибрані зображення елементів ігрового середовища можуть бути творчо доопрацьовані з внесенням додаткових деталей засобами інструмента Canvas із застосуванням функцій ШІ Leonardo.Ai. В такий спосіб для гри «Labyrinth of Darkness» було створено сувій для подальшого відображення завдання для гравця, але йому бракувало певних елементів,

зокрема оригінального штампа у формі черепа, який був доданий авторами в Canvas за допомогою відповідного текстового запита (prompt) (рис. 3).

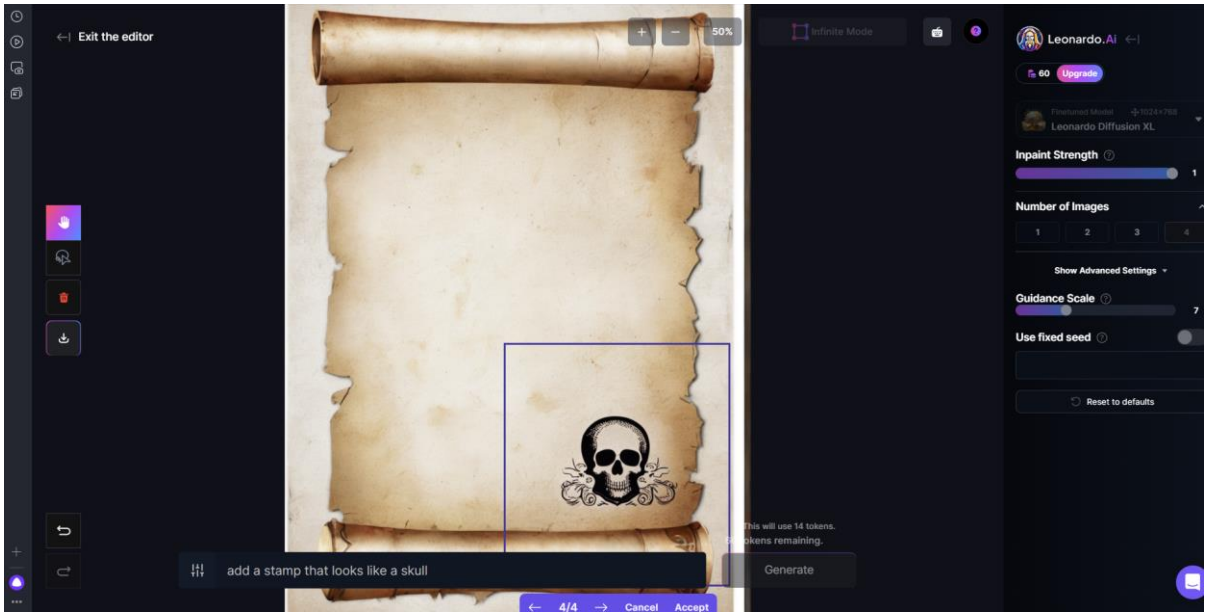


Рис. 3. Результат додавання штампу у вигляді черепу із застосуванням ШІ Leonardo.Ai для гри «Labyrinth of Darkness»

Платформа Leonardo.Ai допомогла трансформувати ескізи в деталізовані ігрові елементи практично в режимі реального часу за допомогою інструмента Realtime Canvas. Також Leonardo.Ai використано для створення HUD (Heads-Up Display) – інтерфейсу, який відображає важливу інформацію під час гри, не відволікаючи увагу гравця від основного процесу. Завдяки цим ШІ-інструментам значно прискорився процес розробки гри «Labyrinth of Darkness», що дозволило забезпечити високий рівень якості візуального контенту. Саме тому використання Leonardo.Ai для створення елементів графічного інтерфейсу не лише оптимізує розробку візуального оформлення, а й сприяє формуванню унікального образу гри, який безпосередньо впливає на її атмосферу та загальне сприйняття користувачем.

2. Інструменти ШІ для створення 3D-моделей у відеоіграх.

У сучасному геймдизайні об’єкти та персонажі визначають структуру візуального середовища гри, а ступінь їхньої деталізації відіграє важливу роль у створенні ефекту занурення для гравця. Розробка високоякісних 3D-моделей вимагає від професійних художників значних часових затрат, у той час як для непрофесійних користувачів оволодіння складними інструментами на кшталт Maya чи Blender часто є складним завданням. Розрив між зростанням вимог до якісного контенту та обмеженими можливостями його створення посилюється ще й високою вартістю та технічною складністю традиційного програмного забезпечення.

Одним з актуальних рішень цієї проблеми є платформа Meshy.ai, яка демонструє, як ШІ може суттєво спростити й прискорити процес створення 3D-моделей та текстур. Сервіс має зручний, інтуїтивно зрозумілий інтерфейс, який дає змогу навіть новачкам створювати базові моделі шляхом завантаження ескізу або введення текстового опису. Алгоритми платформи аналізують вхідні дані, розпізнають геометричну структуру та властивості матеріалів, що забезпечує генерацію реалістичних моделей з урахуванням глибини, текстур й освітлення. Робочий процес на платформі складається з трьох ключових етапів: оброблення текстового запиту за допомогою великих мовних моделей, генерація базової 3D-структури та подальша деталізація й текстурування об’єкта. Серед головних переваг Meshy.ai є здатність обробляти геометрично складні об’єкти й створювати високоякісні, реалістичні текстури. Глибокі нейронні мережі, на яких

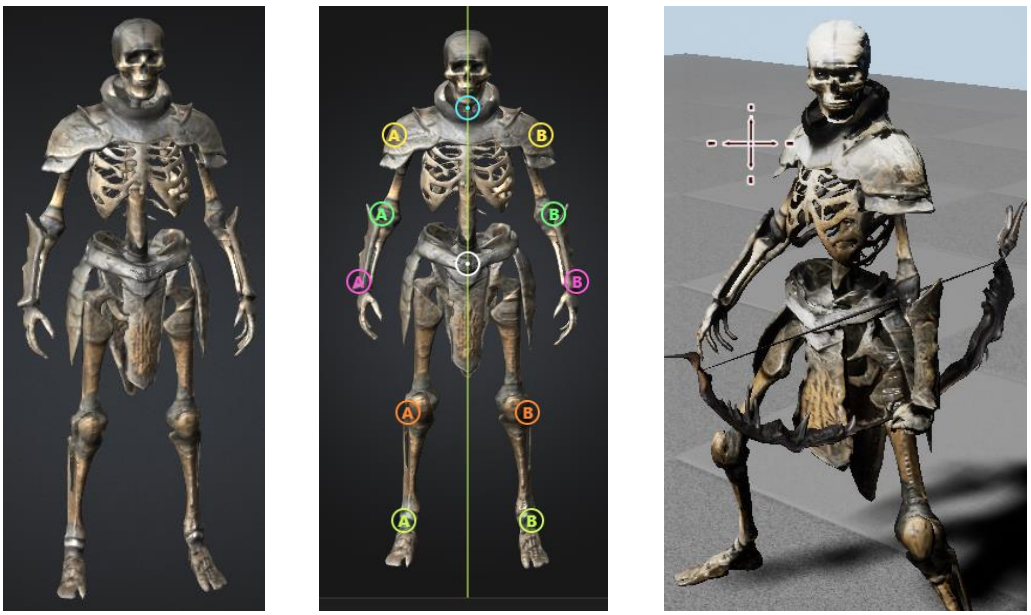
базується сервіс, аналізують великі масиви зразків текстур і генерують нові варіанти, які відповідають сучасним вимогам до графіки. Оптимізований процес генерації дає змогу отримати готові до використання 3D-моделі у форматах .fbx, .obj та .glb з інтегрованими картами текстур [13].

Meshy.ai формує 3D-об'єкти із збалансованим співвідношенням між полігональністю та деталізацією, що забезпечує їхню придатність як для інтер'єрного дизайну (наприклад, меблі, декор), так і для моделювання персонажів, транспортних засобів, органічних структур, архітектурних елементів та інших компонентів віртуального ігрового середовища. Крім того, користувачам надається можливість вибирати з-поміж різних матеріалів – метал, дерево, тканина, скло тощо – і комбінувати їх для досягнення бажаних візуальних характеристик. Сервіс також підтримує генерацію текстур на базі текстових запитів (text-to-texture) та зображень (image-to-texture), що забезпечує гнучкість і зручність у створенні візуального контенту [14].

Створення деталізованих 3D-моделей персонажів, наприклад лицаря-скелета для гри «Labyrinth of Darkness» на рис. 2, за допомогою Meshy.ai здійснювалося на основі завантаженого концептуального зображення. Після автоматизованої генерації 3D-моделі з базовою геометрією та текстурами виконувалася перевірка отриманого об'єкта як безпосередньо у середовищі Meshy.ai, так і у сторонніх професійних 3D-редакторах, як-от: Blender, ZBrush, 3ds Max, Maya. На цьому етапі виконувалося коригування геометрії, усунення дефектів полігональної сітки та ретопологізація, що є критично важливим для подальшої анімації моделі. Додаткове доопрацювання виконували шляхом скульптингу в інструменті ZBrush, де вносили дрібні деталі, зокрема: декоративні елементи на броні, фактуру кісткових поверхонь, тріщини та інші рельєфні особливості.

Наступний етап розробки стосується текстурування, в якому за допомогою інструмента Texture генерується та налаштовується UV-розгортка, а також створюються карти нормалей, рельєфу і глобального освітлення, що поєднуються з базовою дифузною текстурою. Платформа Meshy.ai підтримує генерацію текстур на основі текстових запитів (text prompts), що надає змогу повністю оновлювати візуальне оформлення моделі. У конкретному прикладі для адаптації до стилістики гри було згенеровано текстуру лицарської броні у стародавньому стилі з характерними слідами зношення та декоративними елементами. Текстовий запит (prompt) містив опис зовнішнього вигляду, зокрема потертості на кістках, тріщини та складну орнаментацію на броні, що дозволило системі згенерувати текстуру, наближену до візуальної естетики старовинного ремесла (рис. 4,а). Для покращення візуального сприйняття цього 3D-персонажа гри були застосовані додаткові стилізаційні ефекти, які активуються через функцію Stylize з можливістю налаштування параметрів. Оскільки модель використовуватиметься в інтерактивному середовищі наступним кроком було створення скелетної анімації (rigging). Засобами тієї самої платформи реалізовано автоматизовану функцію Rig, яка забезпечила прив'язку скелетної структури до геометрії моделі (рис. 4,б). Після чого здійснювалося тестування деформацій під час руху та перевірка сумісності з анімаційними сценаріями. Завершальним етапом був експорт моделі у формат, який підтримується ігровими рушіями, для подальшої інтеграції в проєкт (рис. 4,в).

3D-моделі персонажів ігрового середовища, створені за допомогою Meshy (рис. 5), характеризуються високим рівнем деталізації, ретельно опрацьованими текстурами й анатомічно коректною структурою, що забезпечує сумісність і зручність подальшої анімації в ігрових рушіях Unity та Unreal Engine.



а б в

Рис. 4. Поетапне створення 3D-моделі лицаря-скелета для гри «Labyrinth of Darkness» за допомогою ШІ-інструментів:
а) 3D-модель в Meshy.ai з оновленою текстурою й додаванням ефекту металевого покриття;
б) додавання до моделі опорних точок для подальшої анімації в Meshy.ai;
в) застосування розробленої 3D-моделі в ігровому рушії Unreal Engine 4

Рис. 5. 3D-моделі персонажів для гри «Labyrinth of Darkness», розроблених за допомогою ШІ-інструментів: головний герой; ворог «зомбі»; персонаж «бос»; ворог «скелет»; ворог «слиз»

У сукупності наведені можливості визначають Meshy.ai як ефективний інструмент на базі ШІ для розробників відеоігор, дозволяючи їм швидко та ефективно створювати високоякісні 3D-активи без потреби витрат часу на ручне моделювання.

3. Інструменти ШІ для створення музики у відеоіграх.

Аудіосупровід у відеоіграх формує ігрову атмосферу та впливає на емоційне сприйняття гравця. Традиційний процес створення музичного супроводу потребує залучення композиторів, звукозаписних студій та використання вартісного обладнання. Використання сучасних ШІ-рішень, як-от платформи AIVA, оптимізує процес створення музики, робить його доступнішим та технологічно гнучким.

AIVA є інноваційним інструментом на основі штучного інтелекту, який дозволяє генерувати музичні композиції у понад 250 стилях за лічені секунди [15]. Основним

принципом роботи AIVA є використання алгоритмів глибокого навчання. Система аналізує величезні обсяги музичних творів, створених відомими композиторами, та виявляє закономірності в нотних партитурах з метою моделювання стилістичних особливостей. Для уникнення однотипності у композиціях, платформа застосовує глибинні нейронні мережі з метою генерації варіативних музичних фрагментів.

Під час розробки гри «Labyrinth of Darkness» використання платформи AIVA для створення музики допомогло створити унікальне звукове оформлення, яке підкреслює похмуру та напружену атмосферу ігрового простору. Згенеровані штучним інтелектом динамічні, емоційно насичені композиції підсилюють напругу під час дослідження лабіринтів, битв з ворогами та проходження пасток. Музичний супровід динамічно адаптується до розвитку подій, що сприяє глибокому зануренню гравця у віртуальний світ і робить рівень більш захоплюючим та атмосферним.

Платформа AIVA може бути інтегрована в ігрові проєкти задля отримання високоякісних музичних треків без потреби значних витрат часу та ресурсів. Платформа надає інструменти для налаштування та варіативної персоналізації звукового оформлення, що забезпечує відповідність музики конкретному емоційному тону та атмосфері ігрової ситуації.

4. Інструменти ШІ для озвучення персонажів.

Іншим важливим досягненням застосування ШІ в галузі розробки відеоігор є автоматизоване озвучування персонажів. Традиційно процес створення голосового супроводу потребує участі професійних акторів, оренду студій звукозапису та здійснення тривалої постпродакшн-обробки. Однак, завдяки сучасним технологіям на основі ШІ, наприклад, Replica Studios та іншим подібним сервісам, цей процес став значно доступнішим, знизивши витрати та час, необхідний для отримання високоякісного озвучення. Озвучування персонажів за допомогою інструментів ШІ забезпечує високий рівень реалістичності та широкий спектр голосових варіацій, що дозволяє розробникам ігор зберігати повний контроль над аудіосупроводом і наративною складовою [16]. Технологія синтезу мовлення на основі ШІ дозволяє розробникам вибирати з численних голосів, налаштовувати інтонацію, темп та емоційне забарвлення, що дозволяє створювати унікальні озвучки для кожного персонажа гри.

ШІ-технології синтезу мовлення стали значним досягненням для інді-розробників, оскільки дозволяють використовувати готові ШІ-платформи, які мають інтеграційні інструменти та SDK (Software Development Kit), а також відкриті бібліотеки та API-інтерфейси. Це сприяє зниженню фінансових витрат і пришвидшенню впровадження рішень у процес розробки. Наприклад, шведська компанія Neon Giant, яка є розробником популярної кібепанк-екшн RPG «The Ascent», у партнерстві зі Sweet Justice Sound та Altered (розробники Altered Studio) впровадила інноваційний метод озвучування персонажів із використанням ШІ для генерації людських голосів [17]. Завдяки цьому інструменту інтегровано 800 нових діалогових реплік для 40 NPC, які до того функціонували винятково з текстовими субтитрами. Завдяки використанню попередньо записаних голосових даних актора Сема Х'юза згенеровано різноманітні якісні голосові варіанти, які значно покращили ігровий досвід навіть за умов обмежених ресурсів [18].

Крім того, інструменти ШІ суттєво спрощують процес локалізації голосового контенту в іграх. Завдяки сучасним технологіям синхронного машинного перекладу та генерації мовлення розробники можуть автоматизувати переклад і озвучення діалогів персонажів, зберігаючи інтонаційні особливості, емоційний фон і культурні аспекти мови. Це відкриває можливість охопити широку міжнародну аудиторію та забезпечити доступність гри для гравців із різних країн.

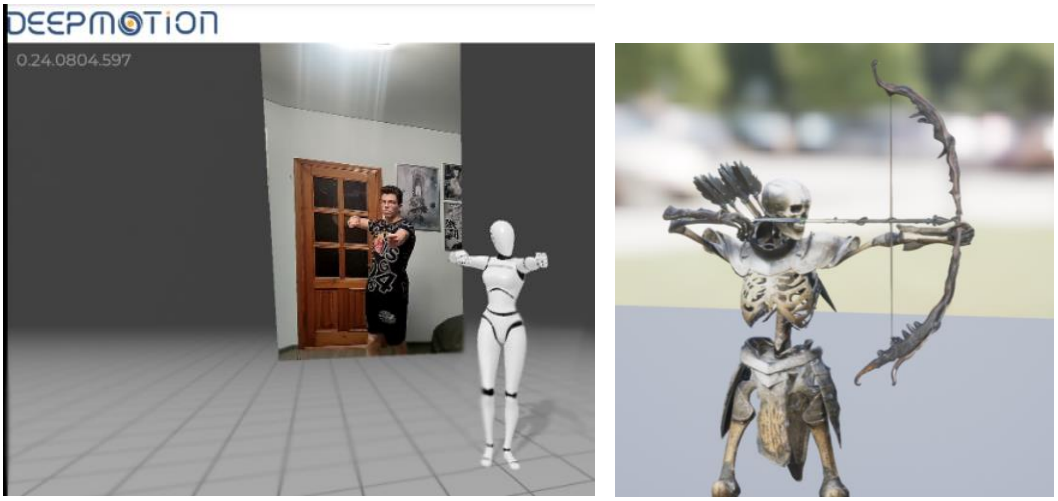
5. Інструменти ШІ для створення анімації у відеоіграх.

Традиційно створення реалістичних анімацій вимагає покадрової ручної роботи. Однак сучасні технології ШІ дозволяють автоматизувати цей процес, застосовуючи алгоритми глибокого навчання для аналізу реальних рухів та їх перенесення на цифрових

персонажів. Одним з основних напрямків використання ШІ в анімації є автоматичне генерація рухів персонажів. Завдяки глибоким нейронним мережам моделі здатні аналізувати відеозаписи з реальними рухами та генерувати плавні анімації. Наприклад, нейронні мережі дають змогу переносити рухи людини на анімованого персонажа, що значно прискорює процес розробки. Ба-більше, ШІ може прогнозувати наступні кадри та генерувати їх, створюючи реалістичні переходи між позами персонажа.

Одним із прикладів успішного застосування ШІ для автоматизованого створення рухів персонажів є платформа DeepMotion (https://www.deepmotion.com). Вона дозволяє завантажити відеозапис із рухами реальної людини, після чого алгоритми ШІ аналізують ці рухи та генерують відповідну 3D-анімацію для персонажа. Технологія спирається на хмарне середовище машинного навчання Motion Intelligence, що забезпечує навчання цифрових персонажів виконанню складних рухових дій, зокрема бойових мистецтв, легкої атлетики, паркуру та танцювальних рухів [19]. Платформа підтримує використання SDK з інтеграцією в ігрові рушії Unity та Unreal Engine, що сприяє її сумісності з типовими робочими процесами розробки відеоігор.

Процес створення анімації починається з того, що користувач завантажує відеозапис рухів реальної людини на вебплатформу DeepMotion (рис. 6,а).



а б

Рис. 6. Аналізу рухів людини (а) та накладання їх на 3D-модель для персонажа гри «Labyrinth of Darkness» за допомогою ШІ від DeepMotion

У проєкті «Labyrinth of Darkness» використовувався відеозапис з рухами натягування лука для генерації анімації скелетного лицаря. Після завантаження відео система автоматично виявляє ключові точки скелета, необхідні для побудови скелетної анімації. Користувачу надається можливість налаштувати додаткові параметри трекінгу, зокрема відстеження рухів голови, рук або запис тільки верхньої частини тіла. На основі цих налаштувань ШІ аналізує рухові дані та генерує 3D-модель скелета із коректною кінематикою, відтворюючи динаміку людини з відеозапису (рис. 6,б).

Результати відображаються у вбудованому 3D-плеєрі, де здійснюється оцінка якості трекінгу та, за потреби, застосовуються інструменти корекції поз або виконується повторна обробка з іншими параметрами. Після отримання задовільного результату анімація експортується у формат, сумісний із цільовим ігровим рушієм.

6. Інструменти ШІ для написання та перевірки коду відеоігор.

Традиційно програмування є трудомістким процесом, який потребує від розробників високої концентрації та точності для виявлення помилок і потенційних вразливостей. Однак сучасні інструменти на основі ШІ здатні автоматизувати численні низку етапів розробки, зокрема оптимізацію коду, генерацію шаблонів і моніторинг дефектів у реальному часі. Наразі одним із найбільш відомих рішень є GitHub Copilot,

який застосовує моделі машинного навчання для аналізу контексту коду та надання рекомендацій із вирішення конкретних завдань [20]. Це сприяє скороченню часу розробки, зниженню кількісті помилок, зумовлених людським фактором, і підвищенню ефективності робочих процесів. Крім того, інтелектуальні системи аналізують код на відповідність кращим практикам програмування, виявляють потенційні вразливості, проблеми безпеки та пропонують шляхи їх усунення вже на ранніх етапах. Такий підхід дозволяє зменшувати час, необхідний для тестування, а розробникам дозволяє зосередитися на творчих аспектах і покращенні взаємодії користувачів із продуктом. Отже, застосування інтелектуальних інструментів підвищує ефективність розробки, а перевірку якості програмного забезпечення робить точнішою, що в кінцевому підсумку забезпечує створення високоякісних і стабільних відеоігор.

Наразі GitHub Copilot використовують понад мільйон розробників, а понад 20 000 організацій вже інтегрували цей інструмент у свої робочі процеси. За оцінками експертів [21] використання GitHub Copilot дозволяє прискорити виконання завдань на 55%, а продуктивність розробників збільшити на 30%, що в перспективі до 2030 року сприятиме зростанню світового ВВП на понад 1,5 трильйона доларів, що забезпечить позитивний вплив на глобальну економіку і сприятиме прискоренню цифрової трансформації.

7. Інструменти ШІ для написання діалогів у відеоіграх.

Розробка сценаріїв та діалогів у відеоіграх традиційно вимагає великих зусиль, оскільки текстова частина потребує одночасної граматичної коректності та узгодженості із сюжетними вимогами. Сучасні моделі обробки природної мови, наприклад GPT, допомагають автоматизувати цей етап, генеруючи варіативні версії діалогів, адаптованих до контексту конкретних ігрових сцен. Це дозволяє сценаристам отримати кілька варіантів тексту та вибрати оптимальний тон і рівень емоційної насиченості.

Інтелектуальні системи аналізують контекст, виявляють логічні невідповідності в сценарії та пропонують виправлення, що скорочує час редагування та покращує якість текстового наповнення. Провідні ігрові студії вже використовують ці технології для генерації квестів, побудови розгалужених сюжетних ліній та автоматичного перекладу діалогів з урахуванням мовно-культурних особливостей. Нейронно-машинний переклад підвищив точність і швидкість оброблення тексту, забезпечивши природні, контекстуально відповідні переклади, що є важливим для збереження цілісності ігрового сюжету [22].

Прикладом успішного використання ШІ для динамічної генерації сюжетів і діалогів є гра AI Dungeon (https://aidungeon.com/) від компанії Latitude. Завдяки мовним моделям GPT-3 ця текстова гра дозволяє гравцям створювати власні сюжетні лінії, які генеруються в режимі реального часу. Інтеграція ШІ надає розробникам інструменти для створення адаптивного ігрового контенту зі складною розгалуженою логікою та локалізацією, що підвищує автентичність і природність взаємодії користувача з грою.

8. Інструменти ШІ для тестування відеоігор.

Сучасні технології ШІ відкривають нові можливості для автоматизації процесу тестування відеоігор, яке є критично важливим для забезпечення якості кінцевого продукту. Традиційні методи ручного тестування вимагають значних людських і часових ресурсів, а також схильні до помилок. Застосування ШІ прискорює виявлення багів та недоліків, автоматизує аналіз тисяч сценаріїв у реальному часі та виявлення прихованих проблем, таких як логічні помилки в геймплеї або аномальні поведінкові збої NPC.

Прикладами такого роду інтегрованих рішень є Razer Game Assistant і Razer QA Companion, які частково інтегровані в систему WYVRN, що оптимізує процес тестування, автоматично виявляючи аномалії і фіксуючи послідовність дій користувача під час тестування [23]. Хмарний плагін AI QA Copilot для рушіїв Unreal Engine і Unity виявляє баги, аналізує проблеми з продуктивністю та генерує звіти про тестування. За результатами досліджень [24] AI QA Copilot виявляє на 20-25% більше помилок, скорочує час тестування на 50% й економить до 40% витрат. Іншим прикладом є використання поведінкових дерев (Behavior Tree) в Unreal Engine 4 (UE4) для

автоматизованого тестування поведінки NPC та ботів. Цей інструмент дозволяє створювати складні реакції персонажів, що оптимізує перевірку ігрової логіки та анімаційних сценаріїв. У грі «Labyrinth of Darkness» використання Behavior Tree забезпечило автоматизовану перевірку реакцій NPC за різних умов, що прискорило виявлення помилок.

Інтеграція ШІ в процес розробки відеоігор змінює підходи до тестування, підвищує їхню ефективність. Однак існують певні технічні та концептуальні обмеження сучасних генеративних моделей, які варто враховувати в контексті подальшого розвитку ігрової індустрії.

Рекомендації щодо використання результатів дослідження щодо покращення безпеки систем обробки даних.

Не існує єдиного цільного інструмента на базі ШІ, здатного охопити різні етапи розробки ігрового процесу через розмаїття контенту та механік. Кожен із сучасних інструментів на базі ШІ орієнтований на виконання певних завдань у межах розробки ігор. Наприклад, Leonardo.Ai та Adobe Substance 3D забезпечують автоматизовану генерацію візуальних елементів та унікальних текстур для ігрових світів, що підвищує гнучкість у розробці, надає можливість швидко адаптувати графіку відповідно до потреб проєкту. Інструменти платформ Meshy.ai та Luma AI здатні генерувати високоякісні, фотореалістичні 3D-моделі на основі текстових описів, а DeepMotion автоматизує створення анімацій персонажів. Такий підхід змінює роль дизайнерів, звільняючи їх від рутинних задач і дозволяючи зосередитися на художньому баченні. Behavior Tree в Unreal Engine 4 є важливим інструментом для створення поведінки NPC за допомогою ієрархічних дерев рішень. Хоча Behavior Tree не є автономною ШІ-системою, він забезпечує створення складних моделей поведінки персонажів. Unity ML-Agents є фреймворком для навчання агентів в ігровому середовищі. Він дозволяє створювати складні поведінкові моделі NPC, використовуючи машинне навчання, що робить персонажів більш реалістичними та адаптивними. Наявність описаних можливостей свідчить про те, що сучасні інструменти ШІ сприяють аналізу поведінки гравців та адаптації ігрових механік на підставі зібраних даних. ШІ-технології вже активно використовуються і в українському сегменті геймдеву, де більшість розробників вважають їх корисними інструментами, а не як новий ризик щодо їх робочих місць.

Застосування інструментів ШІ у процесі створення гри «Labyrinth of Darkness» стало ключовим елементом, який сприяв суттєвому прискоренню розробки та підвищенню якості кінцевого продукту. Інструмент Leonardo.AI став у пригоді для автоматизованої генерації графічного інтерфейсу, який є зручним і стилістично відповідним загальному візуальному оформленню гри. Платформа Meshy.ai забезпечила швидку генерацію високоякісних 3D-моделей, зокрема ворогів і персонажів, що дозволило зосередитися на оптимізації ігрової механіки. Інструмент DeepMotion автоматизував створення реалістичної анімації на основі відеоданих з рухами реальних людей, що надало персонажам живості та динаміки. Платформа AIVA згенерувала музичний супровід, який посилив атмосферу напруги та занурення в ігровий світ. Нарешті, використання ШІ для тестування дозволило оперативно виявити й усунути помилки, покращуючи загальний ігровий досвід. Загалом інтеграція зазначених інструментів сприяла успішній реалізації проекту та формуванню єдиного художнього стилю, в якому поєднуються виразна атмосфера і багаторівневий ігровий процес.

Практична реалізація підтвердила ефективність сучасних інструментів ШІ, які суттєво трансформували всі етапи розробки відеоігор, відкривши нові можливості для оптимізації виробничих процесів та підвищення якості кінцевого продукту. Використання інструментів на базі ШІ дозволило зменшити трудові та часові витрати на розробку і, що не менш важливо, покращити кінцевий результат. Завдяки впровадженню цих інноваційних рішень розробники отримали можливість зосередитись на творчих завданнях, а рутинні операції з генерації 3D-моделей, створення анімацій, написання програмного коду та локалізації текстів наразі стають більш автоматизованими й точними.

Висновки. Генеративний ШІ відкриває нові горизонти в розробці 3D-ігор за рахунок автоматизації моделювання, текстурування та тестування. Використання інструментів на базі ШІ допомагає оптимізувати трудомісткі процеси розробки 3D-моделей та компонентів відеоігор. Це дозволяє ігровим студіям створювати реалістичні віртуальні світи з одночасним скороченням часових і фінансових витрат.

Подальший розвиток інструментів ШІ в геймдеві залежатиме від балансу між технологічним інноваціями й організаційними практиками їх застосування. Ефективна інтеграція ШІ у проекти за умови збереження творчих підходів може сприяти підвищенню рівня якості та інноваційності комп'ютерних ігор.

Список літератури

1. Trofymenko, O., Dyka, A., Loboda, Y., Tolocknov, A., & Bondarenko, M. (2025). Artificial Intelligence in the GameDev. *Electronic Professional Scientific Journal «Cybersecurity: Education, Science, Technique»*, 3(27), 109-119. https://doi.org/10.28925/2663-4023.2025.27.701
2. Yang, D., Kleinman, E., & Harteveld, C. (2024). GPT for Games: An Updated Scoping Review (2020-2024). *arXiv*, 2411.00308. https://doi.org/10.48550/arXiv.2411.00308.
3. Filipović, A. (2023). The role of artificial intelligence in video game development. *Kultura Polisa*, 20(3), 50-67. https://doi.org/10.51738/Kpolisa2023.20.3r.50f
4. Gallotta, R., Todd, G., Zammit, M., Earle, S., Liapis, A., Togelius, J. (2024). Large Language Models and Games: A Survey and Roadmap. *IEEE Transactions on Games*, 13 September 2024, 1-18. https://doi.org/10.1109/TG.2024.3461510.
5. Todd, G., Earle, S., Nasir, M., Green, M., Togelius, J. (2023). Level Generation Through Large Language Models. *Proceedings of the 18th International Conference on the Foundations of Digital Games (FDG '23)*, 70, 1-8. https://doi.org/10.1145/3582437.3587211
6. Anjum, A., Li, Y., Law, N., Charity, M, & Togelius, J. (2024). The Ink Splotch Effect: A Case Study on ChatGPT as a Co-Creative Game Designer. *Proceedings of the 19th International Conference on the Foundations of Digital Games (FDG'24)*, 18, 1-15. https://doi.org/10.1145/3649921.3650010.
7. Li, D., Zhang, S., Sohn, S., Hu, K., Usman, M., & Kapadia, M. (2025). Cardiverse: Harnessing LLMs for Novel Card Game Prototyping. *arXiv*, 2502.07128. https://doi.org/10.48550/arXiv.2502.07128.
8. Cui, Y. (2025). The Exploring of AI Applications in Game Development. *Applied and Computational Engineering*, 158, 59-64.
9. Ling, L., Chen, X., Wen, R., Li, T., & Lc, R. (2024). Sketchar: Supporting Character Design and Illustration Prototyping Using Generative AI. *Proceedings of the ACM on Human-Computer Interaction*, 8, 1-28. https://doi.org/10.1145/3677102.
10. Saersenbek, K. & Chinibayeva, T. (2024). Affective computing methods for simulation of action scenarios in video games. *Procedia Computer Science*, 231, 341-346. https://doi.org/10.1016/j.procs.2023.12.214.
11. Humble, N. (2024). AI Supported Game Development: Exploring Opportunities and Challenges. *2nd Symposium on AI Opportunities and Challenges (SAIOC)*. https://doi.org/10.13140/RG.2.2.30170.76480.
12. AI use in video game development - statistics & facts. https://statista.com/topics/13437/artificial-intelligence-ai-use-in-video-game-development/
13. Meshy AI Review: How I Generated 3D Models in One Minute. https://www.unite.ai/meshy-ai-review/
14. Meshy AI: Transform Text into 3D Textures. https://cutout.pro/learn/meshy-ai/
15. AIVA Technology – Composing Music using AI. https://d3.harvard.edu/platform-digit/submission/aiva-technology-composing-music-using-ai/
16. Replicastudios: AI Voice Actors for Games, Films, and Animation I. https://hunts-

ai.onrender.com/ai/replicastudios/
17. Furkan, Ö. How AI Voice Technology is Transforming the Gaming Industry. https://speaktor.com/ai-voice-in-gaming/
18. Neon Giant teams up with Altered for voice in action RPG game The Ascent. https://www.altered.ai/blog/neon-giant-teams-up-with-altered-for-voice-in-action-rpg-game-the-ascent/
19. Tokarev, K. DeepMotion: AI Driven Motion for Games. https://80.lv/articles/deepmotion-ai-driven-motion-for-games.
20. What is GitHub Copilot? https://docs.github.com/en/copilot/about-github-copilot/what-is-github-copilot.
21. The economic impact of the AI-powered developer lifecycle and lessons from GitHub Copilot. https://github.blog/news-insights/research/the-economic-impact-of-the-ai-powered-developer-lifecycle-and-lessons-from-github-copilot/
22. Game On: How AI is Revolutionizing Game Localization! https://linkedin.com/pulse/game-how-ai-revolutionizing-localization-midlocalize-wq6of
23. Razer Reveals WYVRN: AI-Powered Gaming Ecosystem Set to Transform Play and Development. https://press.razer.com/company-news/razer-reveals-wyvrn-ai-powered-gaming-ecosystem-for-game-developers/
24. Razer launches new Wyvrn game dev platform with automated AI bug tester. https://www.theverge.com/news/632121/razer-wyvrn-developer-ai-qa-gamer-copilot-sensa-haptics-thx.

ARTIFICIAL INTELLIGENCE TOOLS IN THE DEVELOPMENT OF GRAPHIC ELEMENTS, 3D MODELS AND OTHER VIDEO GAME COMPONENTS

O.G. Trofymenko, O.V. Zadereyko, N.I. Loginova, O.V. Shevchenko

National University "Odesa Law Academy"
23, Fontans'ka doroga st., Odesa, 65009, Ukraine
Email: trofymenko@onua.edu.ua

The article is devoted to the study of practical aspects of using various tools based on artificial intelligence (AI) in the development of 3D models and video games. The purpose of the article is to study and systematize the applied aspects of using AI in the development of video games and gameplay, since the existing AI tools have different areas of application and different approaches to content generation, testing and optimization. The research methodology consists in considering possible options for using certain modern AI tools at different stages of developing a 3D game. The work analyzes the impact of AI tools on all stages and aspects of game development using the example of creating the game "Labyrinth of Darkness". The use of AI-based tools, such as Meshy.ai, Leonardo.Ai, AIVA, DeepMotion, GitHub Copilot, Replica Studios and others, made it possible to reduce the complexity of development, reduce time costs and, no less importantly, improve the result. Together, these tools allowed for the effective implementation of the game project and the creation of a holistic, atmospheric game with a unique style and exciting gameplay. Particular attention was paid to exploring the possibilities of optimizing the testing and verification process of the program code using AI. The use of AI became a key element in the process of creating the game “Labyrinth of Darkness”, significantly accelerating development and improving the quality of the final product. Modern AI technologies are radically transforming all stages of video game development, opening up new opportunities for optimizing workflows and improving the quality of the final product. Thanks to the implementation of these innovative solutions, developers were able to focus on creativity, and the routine operations of generating 3D models, creating animations, writing code and localizing texts became more automated and accurate. The introduction of AI tools in game development not only increases the efficiency of production processes but also helps to solve social problems in the industry, in particular, excessive workload on employees (“crunch”).
Keywords: artificial intelligence, AI, AI tools, GameDev, game development, testing, graphic element, 3D modeling, 3D model, graphics, texture, design, animation.

DOI 10.15276/imms.v15.no2.288
УДК 004.056.55

Informatics and Mathematical Methods in Simulation
Vol.15 (2025), No. 2, pp. 288-297

МЕТОДИКА ВИКОНАННЯ ТЕСТІВ НА ПРОНИКНЕННЯ З ВИКОРИСТАННЯМ MITRE ATT&CK

В.В. Яцків[1], Н.Г. Яцків, С.І. Возняк, В.М. Кондратюк

Західноукраїнський національний університет
11, Львівська вул., м. Тернопіль, 46009, Україна
Email: jazkiv@ukr.net[1]

У статті розглянуто структурування тестування на проникнення з використанням фреймворку MITRE ATT&CK. Актуальність дослідження зумовлена потребою у стандартизованому, відтворюваному та реалістичному підході до моделювання дій зловмисників під час оцінки кібербезпеки комп'ютерних систем та мереж. Проведено аналіз сучасних публікацій щодо застосування ATT&CK у тестуванні на проникненні, зокрема відображення активностей на тактики та техніки, сценарного тестування, аналізу прогалин у захисті, використання спеціалізованих інструментів та інтеграції з іншими методологіями (PTES, OSSTMM). Запропоновано методику тестування на проникнення, що охоплює планування, вибір релевантних технік, виконання атак, документування та оцінку ефективності. Апробовано механізми вибору технік ATT&CK на основі профілю об'єкта тестування. Для практичного підтвердження доцільності запропонованого підходу проведено тестування на віртуальній машині Beelzebub:1 (VulnHub), яке продемонструвало повний ланцюг атаки та дозволило охопити 57% технік ATT&CK із відповідної матриці. Аналіз якісних та кількісних показників підтвердив ефективність методики: забезпечено високу повторюваність (90%), зручність документування, оптимізацію часу тесту та виявлення критичних вразливостей. Методика рекомендована для використання у професійній практиці тестування на проникнення та освітніх цілях при викладанні курсу тестування на проникнення комп'ютерних систем. Результатом застосування методики є розуміння потенційних шляхів проникнення, а також чіткі та практичні рекомендації для подальшого посилення захисту досліджуваних комп'ютерних систем. Апробації результатів методики свідчать про її придатність до використання у навчальних курсах, а також у професійній практиці, дозволяючи забезпечити повне, зрозуміле та повторюване покриття усіх етапів процесу тестування на проникнення.

Ключові слова: тестування на проникнення, фреймворк MITRE ATT&CK, методика, тактика, техніка, розвідка.

Вступ. В умовах постійного зростання складності сучасних комп'ютерних систем та розширення поверхні атак, тестування на проникнення (penetration testing) є важливим інструментом оцінки рівня кіберзахисту. Традиційні методики пентесту часто залежать від індивідуального досвіду фахівця, інструментального забезпечення та конкретного контексту системи, що призводить до фрагментарного підходу, відсутності відтворюваності результатів та неповного охоплення потенційних векторів атак. У відповідь на ці виклики все більшої популярності набуває використання фреймворку MITRE ATT&CK – структурованої бази знань про тактики, техніки та процедури (TTPs), які застосовують реальні зловмисники у реальних атаках. ATT&CK забезпечує уніфіковану мову та логічну основу для моделювання дій супротивника, що дає змогу стандартизувати, формалізувати та відтворювати сценарії тестування. Структурування тестів на проникнення за допомогою цього фреймворку дозволяє підвищити якість, прозорість та ефективність тестування, а також забезпечити кращу інтеграцію результатів у систему управління кібербезпекою. Однак на практиці досі залишається невирішеним питання вибору релевантних технік ATT&CK, які є найбільш придатними для певного типу комп'ютерної системи, середовища чи типових загроз. Існує потреба у чітких критеріях, моделях або алгоритмах, що дозволяють обґрунтовано та систематично

обирати техніки тестування. Крім того, бракує методик, які дозволяють гнучко адаптувати процеси тестування на проникнення. до різних сценаріїв, зберігаючи при цьому відповідність стандартам фреймворку MITRE ATT&CK.

Аналіз досліджень і публікацій. MITRE ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) є глобально визнаною базою знань, яка каталогізує тактики, техніки та процедури (TTP) зловмисників на основі реальних спостережень. У 2025 році ATT&CK включає 14 тактик, 205 технік і 468 підтехнік, що охоплюють усі етапи кібератаки – від розвідки до впливу. У тестуванні на проникнення ATT&CK використовується для моделювання поведінки зловмисників, оцінки захисних механізмів і створення детальних звітів. Розглянемо сучасні підходи до використання ATT&CK у пенттесті, включаючи методи, інструменти, переваги, обмеження та інтеграцію з іншими методологіями, такими як PTES.

1. Відображення активностей пенттесту на тактики та техніки ATT&CK. Один із основних підходів полягає у відображенні етапів пенттесту на тактики та техніки ATT&CK, щоб забезпечити всебічне покриття потенційних векторів атак. В роботі [3] описано 11 етапів пенттесту, кожен з яких відповідає тактиці ATT&CK, зокрема:

1) розвідка (Reconnaissance). Використання техніки T1595 (Active Scanning) для збору інформації про мережу за допомогою інструментів, таких як Nmap;
2) початковий доступ (Initial Access). Експлуатація вразливостей веб-додатків (T1190) за допомогою SQLmap для SQL-ін’єкцій;
3) виконання (Execution). Виконання команд через зворотну оболонку (reverse shell) за допомогою Python.
4) стійкість (Persistence). Використання вкрадених cookies через JavaScript;
5) уникнення захисту (Defense Evasion). Обфускація пейлоадів за допомогою Python;
6) доступ до облікових даних (Credential Access). Брутфорс за допомогою Hydra;
7) виявлення (Discovery). Виявлення файлів і директорій за допомогою Python;
8) збір даних (Collection). Автоматизований збір даних із локальних систем;
9) командування та контроль (Command and Control). Використання зворотної оболонки для постійного доступу;
10) ексфільтрація (Exfiltration). Передача даних через зашифровані канали;
11) вплив (Impact). Симуляція атаки шифрувальника.

Цей підхід дозволяє пенттестерам систематично тестувати системи, використовуючи реальні техніки зловмисників, що підвищує реалізм тестів.

2. Сценарії тестування на основі ATT&CK. Сценарне тестування передбачає створення конкретних сценаріїв атак, які базуються на техніках ATT&CK, для досягнення визначених цілей, таких як отримання привілеїв адміністратора домену чи симуляція атаки шифрувальника. В [4] описано методику тестування на основі сценаріїв:

1) визначення цілі (наприклад, доступ до конфіденційних даних);
2) запуск атаки для отримання початкового доступу (наприклад, фішинг через T1566);
3) проведення внутрішньої розвідки (T1016, Network Configuration Discovery);
4) виконання дій, таких як стійкість, переміщення по мережі, ескалація привілеїв;
5) досягнення цілі;
6) оцінка ефективності захисників (Blue Team) і надання рекомендацій.

Цей підхід дозволяє організаціям оцінити свою стійкість до конкретних сценаріїв атак і покращити виявлення та реагування на загрози.

3. Використання ATT&CK як спільної мови. ATT&CK забезпечує стандартизовану таксономію, яка полегшує комунікацію між пенттестерами та захисниками. В [5] зазначено, що пенттестери можуть використовувати техніки ATT&CK у звітах, щоб чітко

описати вразливості та методи їх експлуатації. Це допомагає захисникам зрозуміти, які техніки не виявляються їхніми системами, і розробити відповідні контрзаходи.

4. Виявлення прогалин у захисті. Пенттестери використовують ATT&CK для симуляції технік, щоб перевірити, які з них не виявляються або не запобігаються захисними механізмами організації. Отже, пенттестери можуть виконувати аналіз прогалин, зіставляючи техніки ATT&CK із захисними можливостями організації. Це дозволяє виявити слабкі місця, такі як недостатнє виявлення фішингу чи слабкі конфігурації брандмауерів [6].

5. Інструменти з підтримкою ATT&CK. Пенттестери використовують інструменти, які інтегруються з ATT&CK, для виконання конкретних технік. В [7] описано чотири відкритих інструменти:

– Atomic Red Team. Інструкційний інструмент, який використовує наявні інструменти для виконання технік ATT&CK. Підтримує Windows, Mac, Linux;

– Endgame RTA. Надає 50+ скриптів для атак, працює на Windows;

– Caldera. Серверно-агентська модель для симуляції атак, підтримує Windows 64-bit;

– Metta. Працює на віртуальних машинах, підтримує Windows, Linux, Mac.

Ці інструменти дозволяють пенттестерам симулювати техніки ATT&CK у контрольованому середовищі, надаючи докази ефективності захисних механізмів.

6. Інтеграція з іншими методологіями. ATT&CK часто інтегрується з іншими методологіями пенттесту, такими як PTES (Penetration Testing Execution Standard). Наприклад, поєднання PTES з ATT&CK дозволяє пенттестерам покращити фазу моделювання загроз, симулюючи поведінку просунутих постійних загроз (APTs). Це включає використання ATT&CK для планування сценаріїв і оцінки захисних можливостей [6].

7. Оновлення ATT&CK у 2024 –2025 роках. У 2024 році MITRE ATT&CK v.15 додав підтримку технік, пов’язаних із використанням генеративного ШІ для зловмисних цілей, а також покращив інструменти візуалізації, такі як ATT&CK Navigator. Ці оновлення дозволяють пенттестерам планувати тести, використовуючи сучасні техніки, і оцінювати захист проти нових загроз [1].

ATT&CK часто інтегрується зі стандартом PTES або методологією OSSTMM. На відміну від PTES, яка зосереджена на технічних аспектах, ATT&CK охоплює ширший спектр поведінки зловмисників. OSSTMM пропонує кількісні метрики (RAV), тоді як ATT&CK фокусується на якісному описі технік. Поєднання ATT&CK з PTES дозволяє пенттестерам створювати більш реалістичні сценарії, тоді як OSSTMM краще підходить для комплексної оцінки безпеки.

Застосування ATT&CK в тестуванні на проникнення включє відображення активностей на тактики та техніки, сценарне тестування, використання спільної мови, виявлення прогалин у захисті та застосування інструментів, таких як Atomic Red Team. Інтеграція з PTES і використання інструментів, таких як ATT&CK Navigator, підвищує ефективність тестів. ATT&CK дозволяє пенттестерам створювати реалістичні атаки, оцінювати захист і надавати чіткі рекомендації, але вимагає значних знань і ресурсів.

Мета роботи. Розробити та апробувати методику проведення тестувань на проникнення, яка ґрунтується на використанні фреймворку MITRE ATT&CK, для підвищення ефективності, структурованості та якості процесу пентесту. Для досягнення поставленої мети потрібно вирішити такі завдання:

– провести аналіз сучасного стану методик пентесту, що базуються на ATT&CK;

– запропонувати покрокову методику організації та проведення пентесту із застосуванням фреймворку MITRE ATT&CK;

– створити модель відображення типових атак відповідно до ATT&CK-матриці для конкретних типів інформаційних систем (наприклад, веб-додатки, хмарні сервіси або корпоративні мережі);

– апробувати запропоновану методику шляхом проведення реальних пентестів та оцінити її ефективність за допомогою якісних і кількісних показників.

Методика тестування на проникнення із використанням MITRE ATT&CK. Запропонована методика тестування на проникнення з використанням MITRE ATT&CK містить п’ять кроків.

Крок 1. Планування та розвідка

1.1 Визначення цілей тестування:

– обговорення із замовником цілей та об’єктів тестування;

– формулювання завдання, часові рамки та дозволені дії.

1.2 Попередній збір інформації:

– збір відкритих даних про систему або організацію (наприклад, веб-сайти, соцмережі, публічні репозиторії);

– використання фреймворку ATT&CK для ідентифікації технік: наприклад, T1593 – Search Open Websites/Domains.

Крок 2. Вибір та адаптація релевантних технік (TTPs)

2.1 Вибір технік ATT&CK для моделювання атаки залежно від цілей:

– визначення технік на основі типу об'єкту (мережа, веб-додаток, хмарне середовище);

– врахування специфічних загроз, які характерні для об’єкта тестування (наприклад, відомі вразливості, типові атаки).

2.2 Створення матриці відповідності технік ATT&CK конкретним сценаріям пентесту.

Крок 3. Проведення тестування

3.1 Тестування поділяється на стадії, аналогічні етапам реальної атаки

– початковий доступ. Застосування технік, таких як фішинг T1566, експлуатація веб-сервісів T1190, зовнішніх пристроїв T1200;

– виконання. Використання скриптів, автоматизованих інструментів T1059 - Command and Scripting Interpreter;

– закріплення доступу. Перевірка можливостей збереження доступу після проникнення (наприклад, через реєстр, скрипти автозавантаження);

– підвищення привілеїв. Спроби отримати права адміністратора або системні привілеї (наприклад, T1548 – Abuse Elevation Control Mechanisms);

– уникнення захисних механізмів. Використання методів, що обходять антивіруси та IDS/IPS (наприклад, обфускація коду, виключення логування T1562);

– отримання облікових даних. Випробування методів отримання облікових записів (наприклад, T1555 - Credentials from Password Stores);

– переміщення в мережі. Моделювання технік переміщення між вузлами мережі (наприклад, Pass the Hash T1550, RDP T1021);

– збір та витік інформації. Спроба виявити та вивести чутливі дані (наприклад, стиснення та шифрування для прихованого витоку T1048);

– вплив на роботу систем. Перевірка потенційних способів нанесення шкоди або виведення систем з ладу (наприклад, знищення даних T1485).

Крок 4. Документування результатів

4.1 Відображення використаних технік ATT&CK на конкретні сценарії проникнення.

4.2 Структурований опис знайдених вразливостей з посиланням на ATT&CK ID.

4.3 Підготовка звіту з рекомендаціями щодо усунення виявлених недоліків.

Крок 5. Аналіз та рекомендації

5.1 Визначення рівня покриття технік ATT&CK у тестуванні.

5.2 Надання рекомендацій щодо покращення захисту відповідно до виявлених прогалин (наприклад, пропозиція впровадження нових засобів безпеки або процедур реагування).

Механізми вибору релевантних технік. Для вибору релевантних технік ATT&CK під час пентесту використовуються наступні механізми [8, 9]:

2.1 Аналіз загроз (Threat Intelligence):

– збір інформації про типові атаки на подібні системи чи індустрії;

– вибір технік, що найчастіше використовують зловмисники в конкретних сценаріях.

2.2 Профілювання об’єкта тестування:

– врахування особливостей об’єкта (хмарне середовище, веб-додаток, мобільні пристрої);

– використання технік ATT&CK, що найбільш актуальні для конкретної технології чи середовища.

2.3 Рейтингові та пріоритетні механізми:

– використання пріоритетності технік ATT&CK (наприклад, за ступенем потенційної шкоди або частоти використання у реальних атаках);

– створення спеціальних матриць оцінки, що визначають пріоритетність виконання технік залежно від потенційного впливу на безпеку.

2.4 Автоматизація вибору технік:

– розробка спеціалізованого ПЗ або скриптів, які автоматично рекомендують техніки ATT&CK на основі типових профілів об’єктів тестування;

– використання штучного інтелекту для прогнозування найбільш ефективних технік для конкретного сценарію.

Сценарій тестування на проникнення структурований відповідно до фреймворку MITRE ATT&CK. Для перевірки методики проведення тестування на проникнення використаємо машину Beelzebub: 1 з VulnHub [8].

Цей сценарій може бути використаний як навчальний приклад для студентів з курсу тестування на проникнення.

Етапи тестування [1].

1. Розвідка.

Мета: виявити доступні сервіси та потенційні вектори атаки.

T1595.002 – Active Scanning. Vulnerability Scanning

Використання nmap для сканування відкритих портів:

nmap -p- -sV 192.168.1.55

Виявлено відкриті порти: 22 (SSH) та 80 (HTTP).

T1590.002 – Gather Victim Network Information. Domain Properties.

Використання dirb або dirsearch для виявлення прихованих директорій на веб-сервері:

dirb http://192.168.1.55/

Виявлено /phpmyadmin/ та інші цікаві шляхи.

2. Початковий доступ

Мета: отримати доступ до системи через веб-інтерфейс.

T1190 – Exploit Public-Facing Application

Аналіз HTML-коду сторінки /index.php виявив приховане повідомлення з підказкою: "My heart was encrypted, “beelzebub” somehow hacked and decoded it. -md5"

Генерація MD5-хешу від слова "beelzebub":

echo -n "beelzebub" | md5sum

Отримано хеш: d18e1e22becbd915b45e0e655429d487

$ echo -n "beelzebub" | md5sum

d18e1e22becbd915b45e0e655429d487

Перехід за адресою http://192.168.1.55/d18e1e22becbd915b45e0e655429d487/ відкриває доступ до нових директорій.
T1059.001 – Command and Scripting Interpreter. PowerShell
Аналіз веб-сторінки Talk To VALAK виявив вхідне поле, яке при взаємодії відправляє POST-запит. Перехоплення запиту за допомогою Burp Suite показало наявність пароля в заголовках відповіді: M4k3A****

3. Отримання облікових даних.
Мета: визначити ім'я користувача для використання знайденого пароля.
T1589.002 – Gather Victim Identity Information. Email Addresses.
Використання wpscan для виявлення користувачів WordPress:
wpscan --url http://192.168.1.55/d18e1e22becbd915b45e0e655429d487/--enumerate u
Виявлено користувача: krampus

4. Доступ до облікового запису.
Мета: отримати доступ до системи через SSH.
T1078 – Valid Accounts.
Використання знайдених облікових даних для підключення по SSH:
ssh krampus@192.168.1.55
Пароль: M4k3Ad3a1

5. Підвищення привілеїв.
Мета: отримати права адміністратора (root).
T1055 – Process Injection
Аналіз файлу .bash_history виявив використання експлойту для підвищення привілеїв:
wget https://www.exploit-db.com/download/47009
mv 47009 exploit.c
gcc exploit.c -o exploit
./exploit
Після виконання експлойту отримано доступ з правами root.

6. Уникнення захисту.
Мета: обійти захисні механізми системи.
T1562.001 – Impair Defenses: Disable or Modify Tools.
Під час виконання експлойту можливо було змінено або вимкнено деякі захисні механізми системи для успішного підвищення привілеїв.

7. Витік даних.
Мета: отримати конфіденційні дані з системи.
T1005 – Data from Local System.
Після отримання прав root, доступ до файлів user.txt та root.txt:
cat /home/krampus/user.txt; cat /root/root.txt
Даний сценарій демонструє повний цикл тестування на проникнення, включаючи розвідку, експлуатацію вразливостей, підвищення привілеїв та отримання конфіденційних даних. Структурування дій відповідно до фреймворку MITRE ATT&CK дозволяє систематизувати процес та забезпечити повне покриття можливих векторів атаки.

Примітка. Сценарій призначений виключно для освітніх цілей. Використання описаних методів на системах без відповідного дозволу є незаконним.

Практична апробація методики із використанням MITRE ATT&CK. Під час практичної апробації методики тестування за фреймворком MITRE ATT&CK було проведено низку тестів із використанням різних технік. Основою слугувала віртуальна машина Beelzebub: 1 з VulnHub [8]. Узагальнений тестовий сценарій приведений в таблиці 1.

Таблиця 1.

Узагальнений сценарій тестування

№	Етап ATT&CK	Використані техніки ATT&CK	Результати виконання
1	Розвідка	T1595.002 – Active Scanning, T1590.002 – Domain Properties	Виявлено відкриті порти (22, 80), директорії /phpmyadmin.
2	Початковий доступ	T1190 – Exploit Public-Facing Application, T1059 – Command Interpreter	Отримано доступ до прихованої веб-сторінки, знайдено пароль.
3	Отримання облікових даних	T1589.002 – Gather Identity Information	Встановлено ім'я користувача: krampus.
4	Доступ до облікового запису	T1078 – Valid Accounts	Успішне підключення до системи через SSH.
5	Підвищення привілеїв	T1055 – Process Injection (Kernel Exploit)	Отримано root-доступ через виконання експлойту.
6	Уникнення захисту	T1562.001 – Disable/Modify Security Tools	Успішне обходження систем захисту.
7	Витік даних	T1005 – Data from Local System	Отримано доступ до чутливих даних (файли user.txt, root.txt).

Таким чином, сценарій демонструє повний ланцюжок атаки – від первинної розвідки до повного контролю над системою.

Аналіз ефективності методики. Ефективність запропонованої методики оцінювалася за допомогою якісних та кількісних критеріїв [9 - 11].

Для оцінювання ефективності методики тестування було використано наступні критерії:

1) повнота – здатність методики виявити максимум можливих вразливостей та технік атаки;
2) швидкість – витрати часу на проведення тесту та отримання результатів;
3) повторюваність – здатність інших фахівців повторити сценарій тестування з аналогічними результатами;
4) прозорість – зрозумілість та послідовність документування дій у рамках ATT&CK.

Якісний аналіз ефективності.

1. Повнота. Методика дозволила охопити основні етапи типової атаки, які перелічені в ATT&CK-матриці. Завдяки структурованості, методика сприяла уникненню пропусків важливих етапів або технік.

2. Швидкість. Час на реалізацію атаки значно скоротився завдяки чіткій структурі ATT&CK та визначеним покроковим діям. Часові рамки тестування стали більш прогнозованими.

3. Повторюваність. Студенти, що тестували цю методику, змогли відтворити сценарії з високою точністю (близько 90%). Структурованість ATT&CK дозволила легко контролювати та відстежувати послідовність дій.

4. Прозорість. Методика ATT&CK виявилася зручною для документування результатів. Кожна техніка має чітке визначення та опис, що значно спрощує написання звітів.

До переваги методики необхідно віднести: системний підхід до пентесту, спрощення документування результатів, висока повторюваність та зрозумілість для навчання, можливість автоматизації вибору релевантних технік та TTP. Кількісні результати запропонованої методики наведені в таблиці 2.

Таблиця 2.

Кількісні результати запропонованої методики

Показник ефективності	**Значення**
Загальна кількість використаних технік ATT&CK	8 з 14 (57%)
Час, витрачений на повний цикл атаки	≈ 2 години
Кількість виявлених критичних вразливостей	3
Відсоток успішно повторених дій сторонніми тестувальниками	90%

Разом з тим, методиці притаманні певні обмеження, а саме: 1) вимагає високого рівня знань та попередньої підготовки тестувальників у фреймворку ATT&CK; 2) повне охоплення матриці ATT&CK може бути ресурсномістким для великих та складних систем.

Висновки. Запропонована методика тестування на проникнення з використанням MITRE ATT&CK підтвердила свою ефективність та зручність. Вона дозволяє систематизовано та якісно проводити тести, що сприяє підвищенню ефективності оцінювання безпеки систем.

Результати апробації методики свідчать про її придатність до використання у навчальних курсах, а також у професійній практиці, дозволяючи забезпечити повне, зрозуміле та повторюване покриття усіх етапів процесу тестування на проникнення.

Список літератури

1. MITRE: Mitre att&ck framework. URL: https://attack.mitre.org
2. Al-Sada, Bader, Alireza Sadighian, and Gabriele Oligeri. "Mitre att&ck: State of the art and way forward." ACM Computing Surveys. 2024. No. 57.1. P.1-37
3. Leveraging the MITRE ATT&CK Framework in Penetration Testing of Web Applications. URL: https://www.winmill.com/leveraging-the-mitre-attck-framework/
4. The MITRE ATT&CK framework and scenario-based security testing. URL: https://www.redscan.com/news/mitre-attack-framework-importance-scenario-based-security-testing/
5. Find a Pentesting Provider That Uses the MITRE ATT&CK Framework. URL: https://www.packetlabs.net/posts/find-a-pentesting-provider-that-uses-the-mitre-framework/
6. 5 Penetration Testing Standards to Know in 2025. URL: https://www.sprocketsecurity.com/blog/pentesting-standards-2025
7. 4 open-source Mitre ATT&CK test tools compared. URL: https://www.csoonline.com/article/565132/4-open-source-mitre-attandck-test-tools-compared.html
8. Virtual machines. BEELZEBUB: 1. URL: https://www.vulnhub.com/entry/beelzebub-1,742/
9. Rahman, F. I., Halim, S. M., Singhal, A., Khan, L. Alert: A framework for efficient extraction of attack techniques from cyber threat intelligence reports using active learning. In IFIP Annual Conference on Data and Applications Security and Privacy. 2024. P. 203-220.
10. Abdeen, B., Al-Shaer, E., Singhal, A., Khan, L., Hamlen, K. Smet: Semantic mapping of cve to att&ck and its application to cybersecurity. In IFIP Annual Conference on Data and Applications Security and Privacy. 2023. P. 243–260
11. Li, Z., Zeng, J., Chen, Y., Liang, Z. Attackg: Constructing technique knowledge graph from cyber threat intelligence reports. In: European Symposium on Research in Computer Security. 2022. P. 589–609.

A METHODOLOGY FOR PERFORMING PENETRATION TESTS USING THE MITRE ATT&CK FRAMEWORK

V.V. Yatskiv[1], N.G. Yatskiv, S.I. Vozniak, V.M. Kondratiuk

West Ukrainian National University
11, Lvivska str., Ternopil, 46009, Ukraine
Email: jazkiv@ukr.net[1]

The article examines the structuring of penetration testing using the MITRE ATT&CK framework. The relevance of the research is driven by the need for a standardized, reproducible, and realistic approach to simulating adversary actions during cybersecurity assessments of computer systems and networks. The paper analyzes recent publications on the application of ATT&CK in penetration testing, including mapping activities to tactics and techniques, scenario-based testing, gap analysis, the use of specialized tools, and integration with other methodologies (PTES, OSSTMM). A methodology for penetration testing is proposed, encompassing planning, selection of relevant techniques, execution of attacks, documentation, and effectiveness assessment. Mechanisms for selecting ATT&CK techniques based on the profile of the target system have been tested. To practically validate the proposed approach, testing was conducted on the Beelzebub:1 (VulnHub) virtual machine, demonstrating a full attack chain and covering 57% of the ATT&CK techniques from the corresponding matrix. Analysis of qualitative and quantitative indicators confirmed the effectiveness of the methodology: it ensured high repeatability (90%), convenience of documentation, test time optimization, and identification of critical vulnerabilities. The methodology is recommended for use in professional penetration testing practice as well as for educational purposes in penetration testing courses for computer systems. The application of the methodology results in an understanding of potential intrusion paths and provides clear and practical recommendations for further strengthening the protection of the studied computer systems. The validation of the methodology confirms its suitability for use in both academic courses and professional practice, enabling comprehensive, understandable, and repeatable coverage of all stages of the penetration testing process.

Keywords: Penetration testing, MITRE ATT&CK framework, methodology, tactics, technique, reconnaissance.

ІНФОРМАТИКА ТА МАТЕМАТИЧНІ МЕТОДИ В МОДЕЛЮВАННІ

Том 15, номер 2, 2025. Одеса – 298 с., іл.

INFORMATICS AND MATHEMATICAL METHODS IN SIMULATION

Volume 15, No. 2, 2025. Odesa – 298 p.

Засновник: Національний університет «Одеська політехніка»

Зареєстровано Міністерством юстиції України 04.04.2011р.
Свідоцтво: серія КВ № 17610 - 6460Р

Друкується за рішенням Вченої ради Національного університету «Одеська політехніка», (протокол №13 від 30.06.2025р.)

Адреса редакції: Національний університет «Одеська політехніка»,

1, Шевченка проспект, Одеса 65044 Україна

Web: www.immm.op.edu.ua (immm.opu.ua)

Email: immm.ukraine@gmail.com