

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
МІЖНАРОДНИЙ УНІВЕРСИТЕТ**

**ПЕРЕДОВІ ТЕХНОЛОГІЇ
В ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНІЙ
ІНЖЕНЕРІЇ**

**МАТЕРІАЛИ КОНФЕРЕНЦІЇ
(зимовий форум)**

Одеса, Україна, 14 листопада 2025 р.

**Одеса
2025**

**MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE
INTERNATIONAL UNIVERSITY**

**International Conference “Advanced Technology in
Information and Communication Engineering”
(ATICE’2025)**

Odesa, Ukraine, November 14, 2025

**Conference Proceedings
(winter forum)**

**Odesa
2025**

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
МІЖНАРОДНИЙ УНІВЕРСИТЕТ**

**Міжнародна конференція «Передові технології
в інформаційно-комунікаційній інженерії»
(ПТІКІ'2025)**

Одеса, Україна, 14 листопада 2025 р.

**Матеріали конференції
(зимовий форум)**

**Одеса
2025**

International Conference “Advanced Technology in Information and Communication Engineering” (ATICE'2025): Conference Proceedings (winter forum). Odesa : International university, 2025, 101 p.

DOI:

Міжнародна конференція «Передові технології в інформаційно-комунікаційній інженерії» (ATICE'2025): Матеріали конференції (зимовий форум). Одеса : Міжнародний університет, 2025, 101 с.

Збірка містить праці Міжнародної конференції з інформаційних систем, технологій захисту інформації, використання сучасних інформаційних технологій в управлінні системами за різними галузями народного господарства.

Матеріали публікуються в авторській редакції.

ОРГАНІЗАЦІЙНИЙ КОМІТЕТ

Голова

КІВАЛОВ С.В. – д.т.н., проф., Міжнародний університет, м.Одеса, Україна.

Співголови

СТРЕЛКОВСЬКА І.В. – д.т.н., проф., Міжнародний університет, м.Одеса, Україна.

УРИВСЬКИЙ Л.О. – д.т.н., проф., Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», м.Київ, Україна.

Члени організаційного комітету

БЕРКМАН Л.Н. – д.т.н., проф., Державний університет телекомунікацій, м.Київ, Україна.

КЛИМАШ М.М. – д.т.н., проф., Національний університет «Львівська політехніка», м.Львів, Україна.

ЛЕМЕШКО А. В. – д.т.н., проф., Харківський національний університет радіоелектроніки, м. Харків, Україна.

РОЗЕНВАССЕР Д.М. – к.т.н., доц., Міжнародний університет, м.Одеса, Україна.

РИХЛІК А. – к.т.н., Лодзький технічний університет, м.Лодзь, Польща

СЕМЕНКО А.І. – д.т.н., проф., Національний авіаційний університет, Київ, Україна.

СІМЕНС Е. – д.т.н., проф., Анхальтський університет прикладних наук, м.Кетен, Німеччина.

СОЛОВСЬКА І.М. – к.т.н., доц., Міжнародний університет, м.Одеса, Україна.

СУНДУЧКОВ К.С. – д.т.н., проф., Національний авіаційний університет, м. Київ, Україна.

РЕДАКЦІЙНИЙ КОМІТЕТ

СОЛОВСЬКА І.М. – к.т.н., доц., Міжнародний університет, м.Одеса, Україна.

РОЗЕНВАССЕР Д.М. – к.т.н., доц., Міжнародний університет, м.Одеса, Україна.

КОМІТЕТ МІЖНАРОДНИХ ЗВ'ЯЗКІВ

СТРЕЛКОВСЬКА І.В. – д.т.н., проф., Міжнародний університет, м.Одеса, Україна.

ГЛОБА Л. С. – д.т.н., проф., Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», м.Київ, Україна.

ВЧЕНИЙ СЕКРЕТАР

СОЛОВСЬКА І.М. – к.т.н., доц., Міжнародний університет, м.Одеса, Україна.
ГРИГОР'ЄВА Т.І. – к.т.н., доц., Міжнародний університет, м.Одеса, Україна.

ТЕХНІЧНО-ПРОГРАМНИЙ КОМІТЕТ

СТРЕЛКОВСЬКА І.В. – д.т.н., проф., Міжнародний університет, м.Одеса, Україна.
УРИВСЬКИЙ Л.О. – д.т.н., проф., Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», м.Київ, Україна.
ГЛОБА Л.С. – д.т.н., проф., Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», м.Київ, Україна.

СЕКЦІЇ

СЕКЦІЯ 1. ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

СТРЕЛКОВСЬКА І.В. – д.т.н., проф., Міжнародний університет, м.Одеса, Україна.
ГЛОБА Л.С. – д.т.н., проф., Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», м.Київ, Україна.
ГРИГОР'ЄВА Т.І. – к.т.н., доц., Міжнародний університет, м.Одеса, Україна.

СЕКЦІЯ 2. КОМП'ЮТЕРНІ НАУКИ ТА КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

МИРОШНИК М.А. – д.т.н., проф., Міжнародний університет, м.Одеса, Україна.
СОЛОВСЬКА І.М. – к.т.н., доц., Міжнародний університет, м.Одеса, Україна.
РУСУ О.П. – к.т.н., доц., Міжнародний університет, м.Одеса, Україна.

СЕКЦІЯ 3. КІБЕРБЕЗПЕКА ТА ЗАХИСТ ІНФОРМАЦІЇ

ЛУНТОВСЬКИЙ А.О. – д.т.н., проф., Державна академія Саксонії «Беруфсакадемія», м. Дрезден, Німеччина.
ЙОНА Л.Г. – к.т.н., доц., Міжнародний університет, Одеса, Україна.
МАНЬКО Д.Г. – д.ю.н., проф., Міжнародний університет, м.Одеса, Україна.

СЕКЦІЯ 4. ЕЛЕКТРОННІ КОМУНІКАЦІЇ ТА РАДІОТЕХНІКА

УРИВСЬКИЙ Л.О. – д.т.н., проф., Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», м.Київ, Україна.
ЛЕМЕШКО А.В. – д.т.н., проф., Харківський національний університет радіоелектроніки, м. Харків, Україна.
ПЕДЯШ В.В. – к.т.н., доц., Міжнародний університет, м.Одеса, Україна.

СЕКЦІЯ 5. ІНФОРМАТИКА ТА ПРОГРАМУВАННЯ В ОСВІТІ

КУЧАЙ О.В. - д.п.н., проф., Національний університет біоресурсів і природокористування України.
ЖИГУЛІН О.А. – д.е.н., проф., Міжнародний університет, Одеса, Україна
ГОРБАЧОВ В.Е. – к.т.н., доц., Міжнародний університет, м.Одеса, Україна.

Ковальчук Д.А. Соціальна інженерія як складова гібридних кібероперацій: сучасні тенденції та виклики безпеки.....	72
Григор'єва Т.І., Мельник В.В. Аналіз методів оптимізації безпеки баз даних.....	75

СЕКЦІЯ 4. ЕЛЕКТРОННІ КОМУНІКАЦІЇ ТА РАДІОТЕХНІКА

Стрелковська І.В., Золотухін Р.В., Стрелковська Ю.О. Методика прогнозування кількості абонентів автоматизованої системи управління в низькошвидкісних радіомережах зв'язку..	79
Гінжол В.М., Русу О.П. Шляхи оптимізації масогабаритних показників перетворювачів електричної енергії телекомунікаційного обладнання	83
Грищенко А.О., Педяш В.В. Автоматизоване конфігурування мереж із використанням відкритих програмних засобів	85
Вакарчук А.О., Прядка С.А. Вплив інтерференції на роботу бездротових мереж у щільних міських середовищах	90
Вакарчук А.О., Димов М.В. Дослідження інтелектуальних антенних решіток для адаптивного управління променем у мобільних мережах 5G/6G	93
Вакарчук А.О., Рушчін В.С. Створення системи “Розумний паркінг” з використанням датчиків руху та WI- FI/LORAWAN	95

СЕКЦІЯ 5. ІНФОРМАТИКА ТА ПРОГРАМУВАННЯ В ОСВІТІ

Бельдюгіна С.С. Адаптивне навчання в курсі інформатики та програмування.....	98
--	----

енергетичного рівня індуктивного елемента параметри магнітопроводу повинні задовольняти умову:

$$S_o S_c \geq \frac{IU \Delta t}{J \Delta B} \quad (6)$$

Отримані залежності свідчать, що розміри дроселів пропорційні кількості енергії, яку необхідно передати за один цикл. Тому оптимізація масогабаритних показників індуктивних елементів імпульсних перетворювачів телекомунікаційного обладнання можлива за умов використання матеріалів з вищою магнітною індукцією B , підвищення допустимої густини струму J або збільшення частоти комутації.

Висновки

На основі проведеного аналізу встановлено, що масогабаритні характеристики індуктивних елементів імпульсних перетворювачів визначаються кількістю енергії, яка передається через магнітне коло за цикл роботи. Отримані аналітичні співвідношення (1) – (6) дозволяють оцінити вплив густини струму в обмотках і магнітної індукції в осерді на необхідну площу для розміщення провідників. Мінімізація розмірів можлива при використанні сучасних магнітних матеріалів, оптимізації теплових умов і підвищенні робочої частоти перетворювачів. Зазначені результати можуть бути використані під час проектування високоефективних джерел живлення телекомунікаційних пристроїв.

Література

1. Zehendner M., Ulmann M. Power Topologies Handbook. Texas Instruments, 2016. 216 p.
2. Erickson R. W., Maksimovic D. Fundamentals of Power Electronics. Springer, 2020. 912 p.

УДК 004.738.5

Грищенко А.О., студент,
Педяш В.В., к.т.н., доцент,
Міжнародний університет,
a-gr@ukr.net

АВТОМАТИЗОВАНЕ КОНФІГУРУВАННЯ МЕРЕЖ ІЗ ВИКОРИСТАННЯМ ВІДКРИТИХ ПРОГРАМНИХ ЗАСОБІВ

Анотація. Робота присвячена дослідженню та практичній реалізації системи автоматизованого конфігурування мережевого обладнання на основі відкритих програмних засобів. Розглянуто концепцію *Infrastructure as Code* та її застосування для адміністрування мережевої інфраструктури. Розроблено модульну систему автоматизації на мові Python з використанням бібліотеки *Netmiko* для налаштування лабораторного SIP-стенду на базі обладнання *Cisco*. Проведено порівняльний аналіз ефективності ручного та автоматизованого підходів до конфігурування, який продемонстрував скорочення часу налаштування у 60 разів при підвищенні надійності та відтворюваності конфігурацій.

Сучасні корпоративні мережі характеризуються зростаючою складністю інфраструктури, великою кількістю пристроїв різних виробників та необхідністю оперативного реагування на зміни бізнес-вимог. Традиційний підхід до адміністрування

мережевого обладнання через ручне введення команд стає неефективним та схильним до помилок людського фактора. Це обумовлює потребу у впровадженні автоматизованих систем управління конфігураціями, які забезпечують швидкість, точність та відтворюваність налаштувань [1].

Метою даної роботи є розробка та дослідження системи автоматизованого конфігурування мережевої інфраструктури на основі відкритих програмних засобів із застосуванням концепції Infrastructure as Code.

Концепція Infrastructure as Code передбачає опис інфраструктури у вигляді програмного коду, який може бути версіонований, повторно використаний та автоматично застосований до цільових систем [2]. Цей підхід забезпечує декларативність конфігурацій, їх відтворюваність та можливість аудиту всіх змін. У сфері мережевого адміністрування IaC дозволяє зберігати конфігурації пристроїв у текстових файлах, які легко піддаються версіонуванню в системах контролю версій типу Git, та автоматично розгортати їх на обладнанні різних виробників.

Для практичної реалізації системи автоматизації було обрано мову програмування Python завдяки її простоті, широкій екосистемі бібліотек для роботи з мережевим обладнанням та кросплатформеності [3]. Основним інструментом взаємодії з пристроями виступає бібліотека Netmiko, яка надає високорівневий інтерфейс для роботи з SSH-з'єднаннями та підтримує специфіку командних інтерфейсів різних виробників мережевого обладнання. Архітектура розробленої системи побудована на принципах модульності та відокремлення конфігураційних даних від програмної логіки [4].

Програмна частина системи автоматизації реалізована у вигляді модульного Python-проєкту, архітектура якого побудована з дотриманням принципів Infrastructure as Code та відокремлення конфігураційних даних від програмної логіки. Структура проєкту представлена у таблиці 1, де показано організацію файлів та каталогів з описом призначення кожного елемента системи.

Центральним елементом архітектури є файл `devices.yaml`, який виконує роль декларативного опису інфраструктури та містить параметри підключення до кожного мережевого пристрою, включаючи IP-адресу, облікові дані, тип пристрою для коректної ініціалізації драйвера Netmiko та шлях до файлу конфігурації. Формат YAML обраний через його високу читабельність та зручність ручного редагування без спеціалізованих інструментів. Власне конфігураційні команди для кожного пристрою винесені в окремі текстові файли у каталозі `configs/`, що дозволяє версіонувати їх у системах контролю версій та змінювати конфігурації без втручання в програмний код [5]. Головний скрипт `main.py` реалізує логіку управління процесом автоматизації: зчитування параметрів пристроїв з YAML-файлу, встановлення SSH-з'єднань через контекстний менеджер Netmiko, виконання попередніх ехес-команд при необхідності, застосування основної конфігурації з текстових файлів та збереження змін у енергонезалежну пам'ять пристроїв командою `write memory`.

Таблиця 1 – Структура файлів проєкту автоматизації

Шлях до елемента	Опис призначення
<code>sip_automation/</code>	Кореневий каталог проєкту
<code>main.py</code>	Головний виконавчий скрипт Python з логікою підключення та конфігурування
<code>devices.yaml</code>	Конфігураційний файл з параметрами підключення до пристроїв
<code>configs/</code>	Каталог для текстових файлів конфігурацій окремих пристроїв
<code>configs/cisco_2960.txt</code>	Файл конфігурації комутатора з налаштуваннями VLAN та портів
<code>configs/cisco_2811.txt</code>	Файл конфігурації маршрутизатора з налаштуваннями CME, DHCP та SIP
<code>logs/</code>	Каталог для автоматично створюваних журналів виконання

Лабораторний стенд для апробації розробленої системи включає маршрутизатор Cisco 2811, комутатор Cisco 2960 та два IP-телефони Cisco 7942G. Мережева топологія спроектована з використанням технології віртуальних локальних мереж: VLAN 10 виділено виключно для голосового трафіку IP-телефонів, VLAN 100 — для трафіку управління. Маршрутизатор виконує роль комплексного сервера, що включає функції CME (Cisco Unified Communications Manager Express), DHCP-сервера для автоматичної роздачі IP-адрес, TFTP-сервера для доставки прошивок і конфігураційних файлів телефонам, а також SIP-реєстратора [4]. На маршрутизаторі налаштовано два фізичні інтерфейси: Fa0/0 отримує IP-адресу через DHCP для можливого підключення до Інтернету, а Fa0/1 використовується для внутрішньої комунікації з комутатором через технологію “router-on-a-stick” з підінтерфейсами Fa0/1.10 (192.168.10.1) для VLAN 10 та Fa0/1.100 (192.168.100.1) для VLAN 100. Логічна схема мережевої топології представлена на рисунку 1.

Центральним елементом архітектури є файл `devices.yaml`, який виконує роль декларативного опису інфраструктури та містить параметри підключення до кожного мережевого пристрою, включаючи IP-адресу, облікові дані, тип пристрою для коректної ініціалізації драйвера Netmiko та шлях до файлу конфігурації. Формат YAML обраний через його високу читабельність та зручність ручного редагування без спеціалізованих інструментів. Власне конфігураційні команди для кожного пристрою винесені в окремі текстові файли у каталозі `configs/`, що дозволяє версіювати їх у системах контролю версій та змінювати конфігурації без втручання в програмний код [5]. Головний скрипт `main.py` реалізує логіку управління процесом автоматизації: зчитування параметрів пристроїв з YAML-файлу, встановлення SSH-з'єднань через контекстний менеджер Netmiko, виконання попередніх `exec`-команд при необхідності, застосування основної конфігурації з текстових файлів та збереження змін у енергонезалежну пам'ять пристроїв командою `write memory`.

Програма використовує високорівневий метод `send_config_from_file` бібліотеки Netmiko, який автоматично входить у режим глобальної конфігурації, виконує команди з файлу послідовно та коректно обробляє коментарі і порожні рядки. Модульна архітектура проєкту забезпечує чіткий поділ обов'язків: базова ініціалізація пристроїв з налаштуванням SSH-доступу виконується вручну один раз, а вся подальша функціональна конфігурація застосовується автоматично, що робить процес розгортання стенду швидким, відтворюваним і надійним.

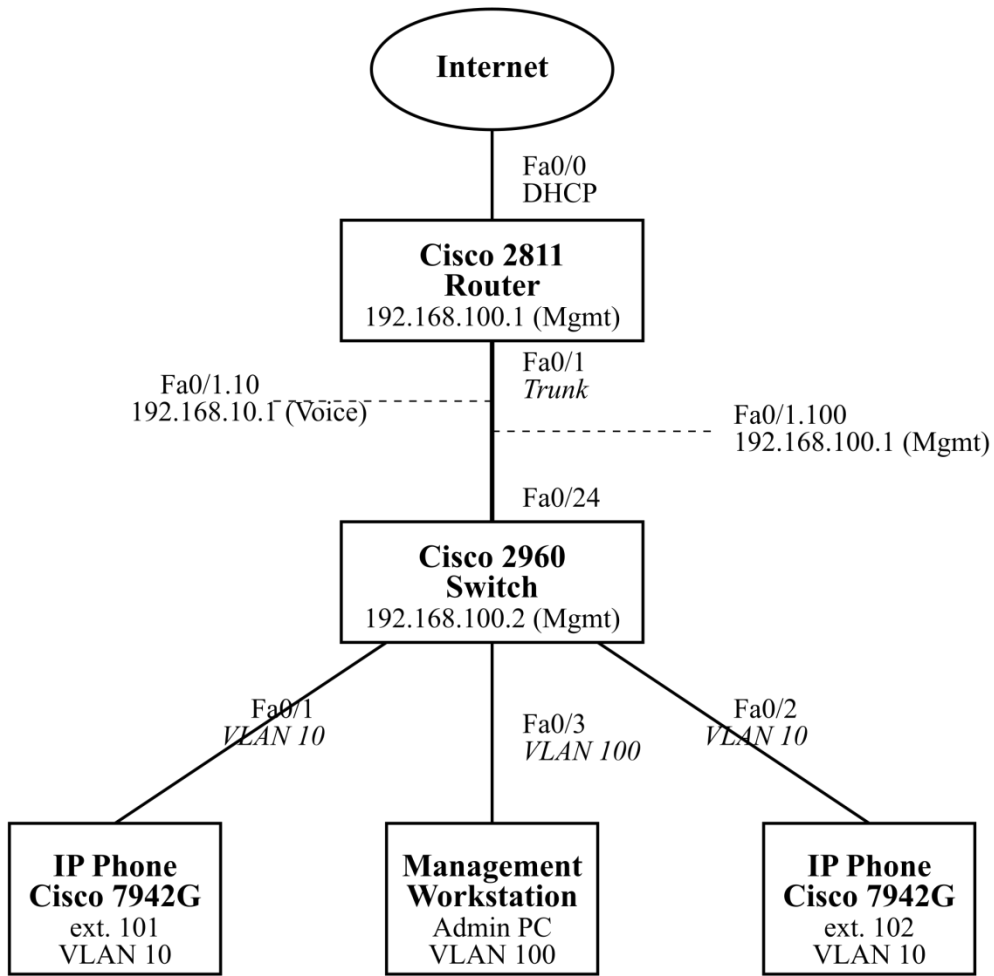


Рисунок 1 – Логічна схема топології лабораторного SIP-стенду

Дослідження ефективності розробленої системи проводилося шляхом порівняльного аналізу часових витрат між ручним та автоматизованим підходами до конфігурування обладнання. Для забезпечення чистоти експерименту обидва пристрої були повністю скинуті до заводських налаштувань, після чого виконувалася базова ініціалізація для забезпечення SSH-доступу. Результати вимірювань представлено у таблиці 2, де показано детальне розбиття загального часу на окремі етапи процесу конфігурування.

Таблиця 2 – Порівняння часових витрат на конфігурування обладнання

Етап процесу	Ручне конфігурування	Автоматизоване конфігурування
Базова ініціалізація пристроїв	15 хв	15 хв
Налаштування комутатора Cisco 2960	8 хв	12 с
Налаштування маршрутизатора Cisco 2811	22 хв	18 с
Перевірка конфігурації та виправлення помилок	12 хв	0 хв
Документування виконаних дій	10 хв	0 хв (автоматичне логування)
Загальний час після базової ініціалізації	52 хв	30 с
Загальний час з урахуванням ініціалізації	67 хв	15 хв 30 с

Аналіз результатів виявляє значне скорочення часу на етапі безпосереднього конфігурування пристроїв з тридцяти хвилин ручної роботи до тридцяти секунд автоматизованого виконання, що становить прискорення у шістдесят разів. Особливо помітною є економія часу на етапі налаштування маршрутизатора, конфігурація якого містить значно більше команд порівняно з комутатором через наявність налаштувань CME, DHCP, TFTP та SIP-сервісів. При ручному конфігуруванні адміністратор витрачає додатковий час на перевірку правильності введення кожної команди та виправлення синтаксичних помилок, тоді як автоматизована система виконує всі команди послідовно з гарантованою коректністю. Документування виконаних дій при ручному підході вимагає окремих зусиль для створення звітів, тоді як розроблена система автоматично веде детальний лог з мітками часу для кожної операції, що створює повний аудит-слід для подальшого аналізу та дотримання вимог комплаєнсу.

Дослідження стійкості системи до помилок включало серію тестів з навмисним внесенням різноманітних проблемних ситуацій: відсутність файлу конфігурації, недоступність пристрою через мережу, невірні облікові дані та синтаксичні помилки у командах. У всіх випадках система коректно перехоплювала виключення, записувала детальну інформацію про помилку в лог-файл з вказівкою IP-адреси проблемного пристрою та продовжувала обробку наступних пристроїв у списку без аварійного завершення. Це критично важливо для масштабних інфраструктур, де тимчасова недоступність одного пристрою не повинна блокувати конфігурування решти обладнання. Важливим результатом дослідження стало підтвердження повної відтворюваності конфігурації: після кожного повного скидання обладнання запуск автоматизаційного скрипта гарантовано відновлював ідентичну робочу конфігурацію з усіма функціональними можливостями, що особливо цінно для навчальних лабораторій та забезпечення швидкого відновлення після інцидентів у корпоративному середовищі.

На основі проведеного дослідження можна зробити наступні висновки. Розроблена система автоматизації конфігурування мережевого обладнання на основі Python та Netmiko забезпечує значне скорочення часових витрат на адміністрування при одночасному підвищенні надійності та відтворюваності конфігурацій. Застосування концепції Infrastructure as Code дозволяє версіонувати конфігурації в системах контролю версій, швидко розгортати ідентичні налаштування та вести повний аудит-слід всіх змін. Модульна архітектура рішення з чітким розділенням виконавчого коду, конфігураційних даних та журналів забезпечує масштабованість та можливість адаптації для різних типів мережевого обладнання. Економія часу адміністратора досягає 60-кратного прискорення порівняно з ручним підходом, що при щотижневому використанні дає економію понад 25 годин робочого часу за академічний рік.

Література

1. Edelman J., Oswalt M., Lowe S. Network Programmability and Automation: Skills for the Next-Generation Network Engineer. 2-ге вид. – Sebastopol (CA) : O'Reilly Media, 2021. – 464 с.
2. Morris K. Infrastructure as Code: Managing Servers in the Cloud. 2-ге вид. – Sebastopol (CA) : O'Reilly Media, 2020. – 368 с.
3. Jorepalli S., Bairy V. Leveraging Network Automation with Python, Terraform, and Ansible to Enhance Security and Operational Efficiency in Large-Scale Networks. – International Journal of Intelligent Systems and Applications in Engineering, Vol. 12, No. 23s, 2024. – С. 2009-2016.
4. Cisco Systems. Cisco Unified Communications Manager Express System Administrator Guide. Release 12.0. – San Jose (CA) : Cisco Systems Inc., 2019. – 892 с.
5. Humble J., Farley D. Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation. – Boston (MA) : Addison-Wesley Professional, 2010. – 512 с.