УДК 004.75 https://doi.org/10.32515/2664-262X.2026.13(44).52-58

О. Б. Гуральник, **О. С. Савенко**, проф., д-р. техн. наук
Хмельницький національний університет, м. Хмельницький, Україна
e-mail: gurualexua@gmail.com, savenko_oleg_st@ukr.net

Формалізований метод оцінювання критичності змін Infrastructure as Code у програмно-керованих мережах

Запропоновано формалізований підхід до оцінювання критичності змін конфігураційних артефактів у програмно-керованих мережах. Критичність визначається як інтегральна функція деградації зв'язності, досяжності критичних сервісів і узгодженості політик безпеки після застосування зміни. Експериментальна перевірка у відтворюваному середовищі підтвердила інтерпретованість інтегральної метрики та її відповідність фактичним функціональним наслідкам IaC-змін.
програмно-керовані мережі, Infrastructure as Code, критичність змін, класифікація

Постановка проблеми. Автоматизація керування корпоративними мережами на основі SDN (Software-Defined Networking) та підходів Infrastructure as Code (IaC) призводить до перенесення мережевих політик і правил пересилання у декларативні конфігураційні артефакти. У таких системах зміна YAML/JSON-опису або сценарію автоматизації безпосередньо впливає на керуючу площину SDN і може змінювати сегментацію, досяжність сервісів та правила доступу [1-3].

На відміну від класичних підходів до аналізу безпеки, що орієнтовані на виявлення відомих вразливостей або атак, у середовищі IaC постає задача оцінювання критичності конкретної конфігураційної зміни до її впровадження. Одна й та сама зміна може бути некритичною в тестовому сегменті та критичною в продуктивному середовищі, що вимагає врахування не лише змісту артефакту, але й контексту його застосування.

Ключовою проблемою є відсутність формалізованого способу пов'язати зміну конфігурації з її впливом на стан мережі та кількісно оцінити наслідки такого впливу до впровадження модифікації. Необхідним є метод, що дозволяє інтерпретовано визначати критичність конфігураційної зміни на основі моделювання переходу стану мережі та застосування формалізованого порогового правила, придатного для інтеграції в автоматизовані процеси розгортання.

Аналіз останніх досліджень і публікацій. У сучасних дослідженнях підкреслюється, що централізована архітектура SDN перетворює контролер на критичну точку атак і помилок конфігурації. Розвиваються методи статичного виявлення небезпечних шаблонів налаштувань, зокрема аналіз потоків керування та даних в системах автоматизації конфігурації, що підвищує точність виявлення вад ціною зростання обчислювальних витрат [4].

У [5] систематизовано загрози безпеці розподілених контролерів SDN у корпоративних SD-WLAN за рівнями архітектури та типами атак (DDoS, компрометація API, порушення синхронізації), а також проаналізовано механізми захисту: автентифікацію, шифрування, протоколи узгодження стану та інтеграцію IDS.
У [6] досліджено вразливості протоколу OpenFlow і експериментально оцінено вплив криптографічного захисту каналу керування на затримку, пропускну здатність і навантаження контрольної площини, що дозволяє оцінити компроміс між безпекою та продуктивністю.

Огляд [7] узагальнює методи виявлення DDoS-атак у SDN із застосуванням машинного, глибинного та федеративного навчання, акцентуючи увагу на проблемах централізації аналізу, затримок та конфіденційності даних. Робота [12] пропонує DL-модель для детекції атак ін'єкції пакетів у контрольній і дата-площинах, демонструючи високу точність за прийнятних накладних витрат. У [13] показано, що використання методів балансування вибірок у поєднанні з ML/DL-класифікаторами суттєво підвищує ефективність виявлення атак у SDN за умов дисбалансу даних.

У сфері IaC дослідження [8] пропонує знаннєво-орієнтований підхід до керованої розробки з формалізацією правил і залежностей між інфраструктурними компонентами для автоматизованої валідації конфігурацій. В огляді [9] систематизовано методи статичного аналізу IaC (синтаксичний, семантичний, перевірка політик, аналіз залежностей) та інструменти для Terraform, Ansible, CloudFormation із урахуванням проблем масштабованості й інтеграції з CI/CD. У [10] розглянуто чинники надійності IaC, типові джерела помилок та засоби підвищення коректності через формалізацію поведінки ресурсів і автоматизоване тестування. Фреймворк DevSecOps у [11] структурує прийняття рішень щодо інтеграції контролів безпеки в CI/CD. Робота [14] демонструє можливості LLM-орієнтованого агентного підходу до автоматизованого виявлення та виправлення вразливостей в IaC, підтверджуючи перспективність підвищення рівня автоматизації та безпеки управління інфраструктурними артефактами.

Постановка завдання. Метою роботи є розроблення методу формалізованого оцінювання критичності IaC-змін у програмно-керованих мережах, який враховує вплив конфігураційної модифікації на зв'язність топології, досяжність критичних сервісів і дотримання політик доступу. Метод має забезпечувати інтерпретоване кількісне вимірювання наслідків зміни на основі формалізованого представлення стану мережі та переходу між станами.

Задача полягає у побудові формальної моделі переходу стану SDN під впливом IaC-зміни та визначенні інтегрального показника, що дозволяє приймати рішення щодо критичності конфігураційної модифікації на основі порогового правила.

Виклад основного матеріалу. Метод оцінювання критичності SDN IaC-змін базується на формалізації впливу конфігураційного артефакту на стан програмно-керованої мережі та формуванні теоретичної основи для побудови апроксимаційного відображення, яке потенційно дозволяє оцінювати вплив без повного розгортання мережевого стану. У межах роботи під IaC-зміною розуміється модифікація декларативного артефакту, що впливає на керуючу площину SDN, зокрема на політики сегментації, правила пересилання або параметри доступу.

Нехай X_{sdn} - множина IaC-артефактів, що змінюють керуючий стан мережі, а $t \in T$ - момент часу застосування зміни. Кожен артефакт $x \in X_{sdn}$ описується атрибутним вектором $a(x,t)$, який включає контентні характеристики конфігурації, структурні параметри її організації, контекст середовища виконання, поведінкові характеристики суб'єкта внесення змін та гарантійні ознаки походження й цілісності. Такий опис дозволяє відокремити власне вплив конфігурації від умов її застосування та забезпечує формальну основу для побудови класифікаційної моделі.

Стан SDN у момент часу t задається як впорядкована пара топологічної та керуючої складових (1):

$$S(t) = \langle G(t), P(t) \rangle, \tag{1}$$

де $G(t) = (V, E(t))$ - орієнтований граф топології з множиною вузлів V та множиною

ребер $E(t)$; $P(t)$ - множина політик і правил пересилання, що визначають дозволену взаємодію між вузлами.

Застосування IaC-зміни формалізується оператором розгортання B (2), унаслідок чого формується новий стан мережі:

$$S'(t) = B(\{x\}, S(t)). \tag{2}$$

Критичність зміни визначається наслідками переходу від стану $S(t)$ до $S'(t)$. Для кількісного опису цього переходу вводиться інтегральна метрика впливу $J(x) \in [0;1]$ (3), яка агрегує нормовані показники деградації зв'язності, досяжності критичних сервісів і узгодженості політик:

$$J(x) = \alpha L_z(x) + \beta L_d(x) + \gamma L_p(x), \tag{3}$$

де $\alpha, \beta, \gamma \in [0;1]$ та $\alpha + \beta + \gamma = 1$. Компонента $L_z(x) \in [0;1]$ відображає нормовану зміну кількості слабкозв'язних компонент топологічного графа після застосування зміни. Компонента $L_d(x) \in [0;1]$ характеризує частку критичних сервісів або вузлів, для яких порушується дозволена досяжність у графі, індукованому політиками. Компонента $L_p(x) \in [0;1]$ визначається як нормована різниця між канонічними наборами правил пересилання до і після застосування зміни, що забезпечує інваріантність до перейменувань і перестановок.

Формування мітки критичності здійснюється пороговим правилом. Зміна вважається критичною, якщо її інтегральний вплив перевищує встановлений рівень ризику (4):

$$y = \begin{cases} 1, J(x) \geq \tau, \\ 0, J(x) < \tau, \end{cases} \tag{4}$$

де $\tau \in [0;1]$ - поріг критичності, що визначається політикою безпеки або статистичним аналізом історичних інцидентів.

Запропонований метод має двокомпонентну структуру. Базовий рівень реалізує формалізоване оцінювання критичності IaC-зміни через моделювання переходу стану мережі та обчислення інтегральної метрики впливу з подальшим застосуванням порогового правила. Цей рівень є самодостатнім і забезпечує інтерпретоване рішення щодо критичності конфігураційної модифікації.

Додатковий, поглиблений рівень методу передбачає побудову апроксимаційної моделі, що відтворює пороговий предикат критичності на основі атрибутного опису зміни без виконання повного оператора розгортання.

Безпосереднє обчислення $J(x)$ для кожної зміни потребує застосування оператора $B(\cdot)$ і аналізу модифікованого стану мережі, що є обчислювально витратним. З метою інтеграції методу в CI/CD-процеси вводиться апроксимаційна модель, яка відтворює порогове рішення без повної симуляції мережевого стану. Формально задача зводиться до наближення предиката критичності за допомогою відображення (5):

$$\hat{y} = M(a(x,t)), \tag{5}$$

де M - класифікаційна модель, що використовує атрибутний опис артефакту.

Модель M інтерпретується як функціональна апроксимація композиції оператора розгортання та інтегральної метрики впливу. У розгорнутому вигляді (6) це відповідає наближенню предиката:

$$M\left(a\left(x,t\right)\right)\approx 1\left[F\left(S\left(t\right),B\left(\left\{x\right\},S\left(t\right)\right)\right)\geq \tau\right], \tag{6}$$

де $F\left(\cdot\right)$ реалізує обчислення нормованих компонент деградації та їх агрегування в $J\left(x\right)$. Надалі використовується скорочений запис (7):

$$M\left(a\left(x,t\right)\right)\approx 1\left[J\left(x\right)\geq \tau\right]. \tag{7}$$

У такій постановці модель машинного навчання не замінює теоретичну метрику, а наближує її результат, забезпечуючи збереження формальної логіки оцінювання критичності за знижених обчислювальних витрат.

Логіка роботи базового рівня методу представлена на рисунку 1. IaC-зміна застосовується до поточного стану мережі, формується модифікований стан, після чого обчислюються нормовані компоненти деградації та інтегральна метрика впливу, що слугує основою для прийняття рішення щодо критичності зміни.

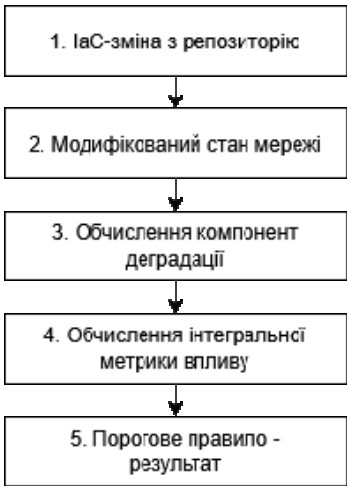

Рисунок 1 – Структура методу оцінювання критичності IaC-змін – базовий рівень
Джерело: розроблено авторами

Експериментальне дослідження ефективності. Для перевірки коректності формалізованої моделі критичності було проведено експериментальне дослідження у відтворюваному програмно-керованому середовищі на базі Mininet, Open vSwitch та контролера Ryu. Стенд реалізовував багатосегментну корпоративну топологію з виділенням критичних сервісів і обмеженням міжсегментної взаємодії відповідно до заданих політик.

На першому етапі дослідження здійснювалася валідація інтегральної метрики впливу (3). Було сформовано набір контрольованих сценаріїв IaC-змін, що включали як нейтральні модифікації конфігурації, так і зміни, які призводили до порушення зв'язності, втрати досяжності критичних сервісів або відкриття несанкціонованого доступу між сегментами. Для кожної зміни x формувався перехід стану мережі $S\left(t\right)\rightarrow S'\left(t\right)=B\left(\left\{x\right\},S\left(t\right)\right)$, після чого обчислювалися компоненти $L_z\left(x\right),L_d\left(x\right),L_p\left(x\right)$ та інтегральний показник (3).

У дослідженні вагові коефіцієнти обрано як $\alpha=0.2,\beta=0.5,\gamma=0.3$, що відображає пріоритет операційної доступності критичних сервісів у корпоративній мережі за одночасного збереження суттєвого внеску дотримання політик сегментації та

доступу. Компонента зв'язності враховується з меншою вагою, оскільки її негативний ефект у більшості практичних сценаріїв проявляється через порушення досяжності сервісів. Порогове значення критичності встановлено як $\tau = 0.5$, що інтерпретується як межа негативного впливу. Мітка критичності формувалася за правилом (4).

Отримані значення для контрольованих сценаріїв наведено в таблиці 1.

Таблиця 1 – Контрольовані сценарії IaC-змін та значення компонент інтегрального впливу

№	Сценарій зміни	$L_z(x)$	$L_d(x)$	$L_p(x)$	$J(x)$	y
1	Перейменування ресурсу без зміни правил	0.0	0.0	0.0	0.00	0
2	Форматування/коментар без зміни політик	0.0	0.0	0.0	0.00	0
3	Дозвіл додаткового порту між сегментами	0.0	0.0	0.4	0.12	0
4	Відкриття міжсегментного доступу (порушення сегментації)	0.0	0.0	0.8	0.24	0
5	Закриття доступу до критичного сервісу	0.0	0.8	0.0	0.40	0
6	Часткова втрата доступності сервісу	0.0	1.0	0.0	0.50	1
7	Ізоляція філії (порушення маршрутизації)	0.6	0.7	0.0	0.47	0
8	Розрив лінка з втратою сервісу	0.7	0.9	0.0	0.59	1
9	Видалення deny-правила для критичного сегмента	0.0	0.0	1.0	0.30	0
10	Одночасне відкриття доступу і зміна сервісу	0.0	0.7	0.8	0.59	1
11	Масштабна зміна ACL для кількох сегментів	0.0	0.6	0.9	0.57	1
12	Втрата доступу до сервісу керування SDN	0.0	1.0	0.5	0.65	1

Джерело: розроблено авторами

Аналіз результатів показує, що нейтральні зміни характеризуються нульовим або близьким до нуля інтегральним впливом, тоді як сценарії, пов'язані з порушенням доступності або суттєвим відхиленням політик, мають значення $J(x)$, що перевищують порогове значення τ. Контрольовані сценарії охоплювали різні механізми деградації мережевих властивостей, що дозволяє оцінити чутливість інтегральної метрики до різних типів порушень. Це підтверджує інтерпретованість та адекватність запропонованої метрики щодо функціональних наслідків IaC-змін.

Попередній аналіз показав збереження відносного порядку критичності сценаріїв при помірній варіації вагових коефіцієнтів, що свідчить про стійкість запропонованої метрики до зміни параметрів агрегування.

У проведеному експерименті інтегральний показник обчислювався шляхом моделювання переходу стану мережі з використанням оператора розгортання та аналізу модифікованої топології. Водночас такий підхід передбачає додаткові обчислювальні витрати, що зумовлює доцільність побудови предиктивних моделей критичності на основі атрибутного опису зміни. У цьому випадку модель $M(a(x,t))$ розглядається як апроксимація порогового предиката критичності.

Висновки. У статті запропоновано формалізований метод оцінювання критичності IaC-змін у програмно-керованих мережах, що базується на моделюванні переходу стану SDN та кількісній оцінці деградації зв'язності, досяжності сервісів і узгодженості політик. Інтегральна метрика забезпечує інтерпретоване відокремлення критичних змін від некритичних на основі порогового правила.

Експериментальна перевірка підтвердила відповідність значень інтегрального показника функціональним наслідкам змін. Запропонований підхід може бути використаний у процесах автоматизованого розгортання конфігурацій для формалізованого оцінювання наслідків конфігураційних змін.

Список літератури

1. Arevalo-Herrera J., Camargo Mendoza J., Martínez Torre J. I., Zona-Ortiz T., Ramirez J. M. Assessing

SDN controller vulnerabilities: A survey on attack typologies, detection mechanisms, controller selection, and dataset application in machine learning. *Wireless Personal Communications*. 2025. Vol. 140, № 1–2. P. 739–775. URL: https://doi.org/10.1007/s11277-025-11748-w (дата звернення: 10.01.2026).
2. Verdet A., Hamdaqa M., Silva L. D., Khomh F. Assessing the adoption of security policies by developers in terraform across different cloud providers. *Empirical Software Engineering*. 2025. Vol. 30, № 3. Art. 74. URL: https://doi.org/10.1007/s10664-024-10610-0 (дата звернення: 10.01.2026).
3. War A., Diallo A., Habib A., Klein J., Bissyandé T. F. Vulnerabilities in infrastructure as code: what, how many, and who? *Empirical Software Engineering*. 2025. Vol. 30, № 5. URL: https://doi.org/10.1007/s10664-025-10672-8 (дата звернення: 10.01.2026).
4. Opdebeeck R., Zerouali A., De Roover C. Control and data flow in security smell detection for infrastructure as code: Is it worth the effort? 2023 IEEE/ACM 20th International Conference on Mining Software Repositories (MSR), Melbourne, Australia. 2023. P. 534–545. URL: https://doi.org/10.1109/msr59073.2023.00079 (дата звернення: 11.01.2026).
5. Shaji N. S., Muthalagu R. Survey on security aspects of distributed software-defined networking controllers in an enterprise SD-WLAN. *Digital Communications and Networks*. 2023. URL: https://doi.org/10.1016/j.dcan.2023.09.004 (дата звернення: 11.01.2026).
6. Riggs H., Khalid A., Sarwat A. I. An Overview of SDN Issues—A Case Study and Performance Evaluation of a Secure OpenFlow Protocol Implementation. *Electronics*. 2025. Vol. 14, № 16. Art. 3244. URL: https://doi.org/10.3390/electronics14163244 (дата звернення: 11.01.2026).
7. Batool S., Aslam M., Akpokodje E., Jilani S. F. A Comprehensive Review of DDoS Detection and Mitigation in SDN Environments: Machine Learning, Deep Learning, and Federated Learning Perspectives. *Electronics*. 2025. Vol. 14, № 21. Art. 4222. URL: https://doi.org/10.3390/electronics14214222 (дата звернення: 17.01.2026).
8. Vasileiou Z., Kumara I., Meditskos G., Tokmakov K., Radolović D., Cruz J. G., Vrochidis S. A knowledge-based approach for guided development of Infrastructure as Code. *Software and Systems Modeling*. 2025. URL: https://doi.org/10.1007/s10270-025-01294-1 (дата звернення: 17.01.2026).
9. Chiari M., De Pascalis M., Pradella M. Static analysis of infrastructure as code: A survey. 2022 IEEE 19th International Conference on Software Architecture Companion (ICSA-C), Honolulu, HI, USA. 2022. URL: https://doi.org/10.1109/icsa-c54293.2022.00049 (дата звернення: 17.01.2026).
10. Sokolowski D., Salvaneschi G. Towards Reliable Infrastructure as Code. 2023 IEEE 20th International Conference on Software Architecture Companion (ICSA-C), L'Aquila, Italy. 2023. URL: https://doi.org/10.1109/icsa-c57050.2023.00072 (дата звернення: 18.01.2026).
11. Akbar M. A., Smolander K., Mahmood S., Alsanad A. Toward successful DevSecOps in software development organizations: A decision-making framework. *Information and Software Technology*. 2022. Vol. 147. Art. 106894. URL: https://doi.org/10.1016/j.infsof.2022.106894 (дата звернення: 18.01.2026).
12. Phu A. T., Li B., Ullah F., Ul Huque T., Naha R., Babar M. A., Nguyen H. Defending SDN against packet injection attacks using deep learning. *Computer Networks*. 2023. Vol. 234. Art. 109935. URL: https://doi.org/10.1016/j.comnet.2023.109935 (дата звернення: 18.01.2026).
13. Bajenaid A., Khemakhem M., Eassa F. E., Bourennani F., Qurashi J. M., Alsulami A. A., Alturki B. Towards Robust SDN Security: A Comparative Analysis of Oversampling Techniques with ML and DL Classifiers. *Electronics*. 2025. Vol. 14, № 5. Art. 995. URL: https://doi.org/10.3390/electronics14050995 (дата звернення: 18.01.2026).
14. Toprani D., Madisetti V. K. LLM agentic workflow for automated vulnerability detection and remediation in infrastructure-as-code. *IEEE Access: Practical Innovations, Open Solutions*. 2025. Vol. 13. P. 69175–69181. URL: https://doi.org/10.1109/access.2025.3560911 (дата звернення: 18.01.2026).

References

1. Arevalo-Herrera, J., Camargo Mendoza, J., Martínez Torre, J. I., Zona-Ortiz, T., & Ramirez, J. M. (2025). Assessing SDN controller vulnerabilities: A survey on attack typologies, detection mechanisms, controller selection, and dataset application in machine learning. *Wireless Personal Communications*, *140*(1–2), 739–775. https://doi.org/10.1007/s11277-025-11748-w
2. Verdet, A., Hamdaqa, M., Silva, L. D., & Khomh, F. (2025). Assessing the adoption of security policies by developers in terraform across different cloud providers. *Empirical Software Engineer*, *30*(3), 74. https://doi.org/10.1007/s10664-024-10610-0
3. War, A., Diallo, A., Habib, A., Klein, J., & Bissyandé, T. F. (2025). Vulnerabilities in infrastructure as code: what, how many, and who? *Empirical Software Engineer*, *30*(5). https://doi.org/10.1007/s10664-025-10672-8
4. Opdebeeck, R., Zerouali, A., & De Roover, C. (2023, May). Control and data flow in security smell detection for infrastructure as code: Is it worth the effort? *2023 IEEE/ACM 20th International Conference on Mining Software Repositories (MSR)*, 534–545. Presented at the 2023 IEEE/ACM 20th International Conference on Mining Software Repositories (MSR), Melbourne, Australia. https://doi.org/10.1109/msr59073.2023.00079

5. Shaji, N. S., & Muthalagu, R. (2023). Survey on security aspects of distributed software-defined networking controllers in an enterprise SD-WLAN. *Digital Communications and Networks*. https://doi.org/10.1016/j.dcan.2023.09.004
6. Riggs, H., Khalid, A., & Sarwat, A. I. (2025). An Overview of SDN Issues—A Case Study and Performance Evaluation of a Secure OpenFlow Protocol Implementation. *Electronics*, *14*(16), 3244. https://doi.org/10.3390/electronics14163244
7. Batool, S., Aslam, M., Akpokodje, E., & Jilani, S. F. (2025). A Comprehensive Review of DDoS Detection and Mitigation in SDN Environments: Machine Learning, Deep Learning, and Federated Learning Perspectives. *Electronics*, *14*(21), 4222. https://doi.org/10.3390/electronics14214222
8. Vasileiou, Z., Kumara, I., Meditskos, G., Tokmakov, K., Radolović, D., Cruz, J. G., … Vrochidis, S. (2025). A knowledge-based approach for guided development of Infrastructure as Code. *Software and Systems Modeling*. https://doi.org/10.1007/s10270-025-01294-1
9. Chiari, M., De Pascalis, M., & Pradella, M. (2022, March). Static analysis of infrastructure as code: A survey. *2022 IEEE 19th International Conference on Software Architecture Companion (ICSA-C)*. Presented at the 2022 IEEE 19th International Conference on Software Architecture Companion (ICSA-C), Honolulu, HI, USA. https://doi.org/10.1109/icsa-c54293.2022.00049
10. Sokolowski, D., & Salvaneschi, G. (2023, March). Towards Reliable Infrastructure as Code. *2023 IEEE 20th International Conference on Software Architecture Companion (ICSA-C)*. Presented at the 2023 IEEE 20th International Conference on Software Architecture Companion (ICSA-C), L'Aquila, Italy. https://doi.org/10.1109/icsa-c57050.2023.00072
11. Akbar, M. A., Smolander, K., Mahmood, S., & Alsanad, A. (2022). Toward successful DevSecOps in software development organizations: A decision-making framework. *Information and Software Technology*, *147*(106894), 106894. https://doi.org/10.1016/j.infsof.2022.106894
12. Phu, A. T., Li, B., Ullah, F., Ul Huque, T., Naha, R., Babar, M. A., & Nguyen, H. (2023). Defending SDN against packet injection attacks using deep learning. *Computer Networks*, *234*(109935), 109935. https://doi.org/10.1016/j.comnet.2023.109935
13. Bajenaid, A., Khemakhem, M., Eassa, F. E., Bourennani, F., Qurashi, J. M., Alsulami, A. A., & Alturki, B. (2025). Towards Robust SDN Security: A Comparative Analysis of Oversampling Techniques with ML and DL Classifiers. *Electronics*, *14*(5), 995. https://doi.org/10.3390/electronics14050995
14. Toprani, D., & Madisetti, V. K. (2025). LLM agentic workflow for automated vulnerability detection and remediation in infrastructure-as-code. *IEEE Access: Practical Innovations, Open Solutions*, *13*, 69175–69181. https://doi.org/10.1109/access.2025.3560911

Oleksandr Huralnyk, Oleg Savenko, Prof., Dr. Tech. Sci.
Khmelnytskyi National University, Khmelnytskyi, Ukraine

A Formalized Method for Assessing the Criticality of Infrastructure as Code Changes in Software-defined Networks

The aim of the article is to develop a formalized method for assessing the criticality of changes to Infrastructure as Code configuration artifacts in software-defined networks. The relevance of the research is determined by the increasing automation of network management processes and the need for early identification of configuration changes that may compromise functional stability or violate security policies in corporate infrastructures. The study focuses on constructing an interpretable model that enables quantitative evaluation of the consequences of configuration modifications prior to their deployment in a production environment.

The paper introduces a formal representation of the state of a software-defined network and models the transition between states under the influence of a configuration change. Criticality is defined as an integral function that aggregates the degradation of topology connectivity, critical service reachability, and access policy consistency. For each change, the network state transition is explicitly modeled, normalized impact components are computed, and an aggregated indicator is formed using weighted coefficients. A threshold-based decision rule is applied to distinguish critical from non-critical modifications. The proposed metric is experimentally validated on controlled scenarios that include neutral, partially degradative, and critical configuration changes. The obtained results confirm the consistency of the integral indicator with the actual functional consequences observed in the modified network state.

The results demonstrate the interpretability and practical applicability of the proposed formalization for structured assessment of configuration changes in automated deployment workflows. The integral metric provides a transparent mechanism for separating critical modifications from non-critical ones based on measurable degradation characteristics.

software-defined networks, Infrastructure as Code, change criticality, classification

Одержано (Received) 19.02.2026 *Прорецензовано (Reviewed) 27.02.2026*
Прийнято до друку (Approved) 12.03.2026