

УДК 004

**В.Павлюхін, магістр гр. КІ-24М,***Центральноукраїнський національний технічний університет*

## ДОСЛІДЖЕННЯ ТА ПРИНЦИПИ ПОБУДОВИ СИСТЕМИ КОНТЕЙНЕРНОЇ ВІРТУАЛІЗАЦІЇ МЕРЕЖЕВОЇ ІНФРАСТРУКТУРИ ДЛЯ ХОСТИНГУ

У статті розроблено програмне забезпечення, яке призначено для системи контейнерної віртуалізації мережевої інфраструктури для хостингу. Метою розробки є дослідження та принципи побудови системи контейнерної віртуалізації мережевої інфраструктури для хостингу. Об'єктом дослідження є процес контейнерної віртуалізації мережевої інфраструктури для хостингу. Предметом дослідження є методи контейнерної віртуалізації мережевої інфраструктури для хостингу. Методи дослідження базуються на методах побудови комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи контейнерної віртуалізації мережевої інфраструктури для хостингу. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

**контейнерна віртуалізація, мережева інфраструктура, хостинг**

**Постановка проблеми.** З появою контейнеризації 10 років тому ми побачили компактний, зручний та портативний спосіб запуску програм безпосередньо одночасно з віртуалізацією. Основна відмінність полягає в архітектурі. Контейнери мають те саме ядро, що й гостьова система, і тому не віртуалізують низькорівневі компоненти, такі як центральний процесор (ЦП).

З одного боку, вони легші та гнучкіші, ніж віртуальні машини (VM). З іншого боку, VM можуть точніше задовольняти потреби низькорівневих компонентів і є повністю автономними системами. Яка архітектура найкраще підходить для розробки програми сьогодні?

У даній магістерській роботі ми вивчимо два основні віртуальні методи розгортання. Ми порівняємо обидва методи за кількома критеріями: сумісність на основі користувацького досвіду та простота встановлення/розгортання, масштабованість на основі автоматичної еластичності робочого навантаження та енергоефективність з точки зору енергії та комп'ютерних ресурсів. Після тестів ми розробимо програмну реалізацію системи контейнерної віртуалізації мережевої інфраструктури для хостингу.

**Аналіз останніх досліджень і публікацій.** При аналізі останніх досліджень і публікацій [1-30] було виявлено певні прогалини у забезпеченні системи контейнерної віртуалізації мережевої інфраструктури для хостингу.

**Мета й завдання дослідження.** Метою роботи є дослідження та принципи побудови системи контейнерної віртуалізації мережевої інфраструктури для хостингу.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

- Огляд існуючих систем контейнерної віртуалізації мережевої інфраструктури для хостингу.
- Дослідження системи контейнерної віртуалізації мережевої інфраструктури для хостингу.
- Програмна реалізація системи контейнерної віртуалізації мережевої інфраструктури для хостингу.

Об'єктом дослідження є процес контейнерної віртуалізації мережевої інфраструктури для хостингу.

Предметом дослідження є методи контейнерної віртуалізації мережевої інфраструктури для хостингу.

Методи дослідження базуються на методах побудови комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.

**Виклад основного матеріалу.** Віртуалізація – це процес використання програмного забезпечення для створення віртуального ресурсу, який працює на окремому рівні від фізичного обладнання. Найпоширенішим випадком використання віртуалізації є хмарні обчислення.

Ви можете запускати кілька віртуальних машин на одному комп'ютері за допомогою віртуалізації. Ці віртуальні машини є незалежними системами, але мають спільну фізичну ІТ-інфраструктуру та керуються гіпервізором.

Віртуалізація набула величезного значення в останній галузі програмного забезпечення. Очікується, що до 2026 року світовий ринок віртуалізації додатків становитиме 5,76 мільярда доларів. Це пояснюється тим, що віртуалізація дозволяє користувачам отримувати доступ до додатків і функцій без їх встановлення на комп'ютер.

Ця хмарна технологія економить гроші, час і місце для зберігання, пропонуючи всі можливості хмарних обчислень. Від неї вигоду отримують як великі підприємства, так і малий бізнес. Деякі з переваг віртуалізації:

- Доступність усіх ресурсів ОС для програм.
- Добре налагоджена функціональність.
- Кращі інструменти та засоби контролю безпеки.
- Надійні інструменти управління.
- Економія коштів та висока ефективність.
- Централізоване робоче навантаження без накладних витрат.

VirtualBox, VMware Workstation Player та Microsoft Hyper-V є найпопулярнішими постачальниками віртуальних машин.

#### **Обмеження віртуалізації**

Віртуалізація – це потужна технологія, яка має багато переваг, але також має кілька обмежень, про які слід знати. Деякі з основних обмежень віртуалізації включають:

- Для роботи кількох операційних систем та віртуальної копії обладнання потрібні значні ресурси оперативної пам'яті та процесора.
- Перехід між приватними та публічними хмарами та центрами обробки даних ускладнює життєвий цикл розробки програмного забезпечення.
- Він монолітний і запускає програми як окремі великі файли.
- Додає обчислювальні ресурси та дуже швидко циклічно перезавантажується/завантажується.
- Він не запускає деякі програми належним чином.
- Деякі старіші або спеціалізовані програми можуть бути несумісними з програмним забезпеченням для віртуалізації або можуть вимагати додаткового налаштування для належної роботи.
- Віртуалізовані середовища можуть бути не такими простими в масштабуванні, як фізичні, особливо коли йдеться про додавання додаткових апаратних ресурсів.

Контейнери – це засіб ізоляції програми від її оточення шляхом інкапсуляції її залежностей та конфігурацій в одному блоці. Після цього цей блок можна перенести в інші середовища, такі як приватні хмари, публічні хмари та центри обробки даних.

Контейнери є легшими та гнучкішими, коли справа доходить до віртуалізації вашого середовища без гіпервізора. Вони дозволяють DevOps зосередитися на розробці та розгортанні коду, що забезпечує швидше надання ресурсів. Контейнеризований додаток поводить себе послідовно в середовищах розробки, тестування та виробництва.

Як згадувалося раніше, контейнеризація – це процес пакування кожного компонента, необхідного для запуску програми або мікросервісу, включаючи пов'язані бібліотеки. Кожен

контейнер складається з коду, залежностей та самої ОС. Це дозволяє програмам працювати однаково на різних платформах.

Контейнеризація – це форма віртуалізації ОС, яка використовує можливості операційної системи для ізоляції процесів та контролю їх доступу до пам'яті, дискового простору та процесорів.

Поширення контейнеризації розпочалося з Docker, платформи з відкритим кодом для створення, розгортання та управління контейнерними додатками. З її появою у 2013 році технологія та екосистема контейнерів зазнали значного розвитку.

Деякі переваги контейнеризації:

- Зменшення завантаженості ресурсів управління ІТ.
- Вимоги до меншого розміру.
- Швидше завантаження та спрощені оновлення безпеки.
- Менше коду для міграції, перенесення або завантаження робочих навантажень.
- Швидша доставка.
- Легше управління.

Контейнеризація працює шляхом спільного використання ядра хост-ОС з іншими контейнерами як ресурсу лише для читання. Ви можете розгорнути кілька контейнерів на одному сервері або віртуальній машині, оскільки вони легкі та масштабовані.

Таким чином, ви підтримуєте лише одну ОС і не присвячуєте цілий сервер одній програмі. Контейнеризація є відповіддю на кілька проблем DevOps. Саме тому багато підприємств застосовують цей підхід для міграції керованих сервісів у хмару.

Контейнери дозволяють розбивати програми на найменші компоненти або мікросервіси. Ці сервіси розробляються та розгортаються незалежно, що усуває монолітність.

Наприклад, якщо ви підтримуєте кілька кнопок дій на своєму веб-сайті, збій однієї з них не впливає на продуктивність інших. Це зменшує час простою, навантаження на обслуговування та залежність.

### **Обмеження контейнеризації**

Як і віртуальні машини, контейнери також мають деякі обмеження.

- Усі контейнери повинні працювати на схожих операційних системах.
- Якщо контейнери базуються на іншій ОС, їм потрібен інший хост.
- Вони можуть створювати вразливості безпеки в ядрі ОС, оскільки всі контейнери на хост-машині використовують цю ОС.
- Це рішення все ще розробляється та вдосконалюється, тому його впровадження може бути складнішим.

Контейнеризація та віртуалізація мають свої сильні та слабкі сторони. Вони використовуються незалежно для задоволення потреб вашого бізнесу. Їх також можна використовувати разом для створення ефективної ІТ-інфраструктури для DevOps.

### **1. Ізоляція**

Контейнери та віртуальні машини забезпечують різні ступені ізоляції. Контейнерній системі потрібна базова ОС для надання базових послуг усім контейнеризованим програмам. З іншого боку, гіпервізор запускає віртуальні машини зі своєю ОС, яка використовує апаратну підтримку.

В результаті, контейнерні системи мають менші накладні витрати, ніж віртуальні машини, і зазвичай орієнтовані на середовища з тисячами контейнерів. Вони забезпечують ізоляцію сервісів між контейнерами, тоді як віртуальна машина забезпечує повністю ізольоване середовище. Це призводить до обмеженого доступу до ресурсів для контейнерних сервісів, таких як файлові системи.

Щоб поєднати можливості контейнеризації та віртуалізації, можна обрати паравіртуалізацію. Вона ізолює програми завдяки підтримці віртуальної пам'яті та вимагає спеціальних драйверів пристроїв у віртуальній машині, підключеній до ОС через гіпервізор.

Простіше кажучи, віртуальні машини, якими керують гіпервізори, використовують обладнання віртуальної машини, тоді як контейнерні системи надають служби ОС від базового хоста для ізоляції програм за допомогою обладнання віртуальної пам'яті.

## **2. Операційна система**

Контейнери та віртуальні машини суттєво відрізняються з точки зору автономності ОС. Контейнери дотримуються віртуалізації ОС, що означає, що вони використовують ресурси хост-ОС. З іншого боку, кожен екземпляр у віртуальній машині є повноцінною гостьовою ОС сам по собі.

Таким чином, віртуальна машина містить гостьову ОС, віртуальну копію обладнання для її запуску, програму, її бібліотеки та залежності. Вона може запускати різні операційні системи на одному фізичному сервері. Оскільки контейнери віртуалізують ОС, вони містять лише програму з її бібліотеками та залежностями.

Оскільки екземпляр віртуальної машини віртуалізує всю ОС, він зрештою додає компоненти, не пов'язані з вашою програмою. Однак, незалежно від типу операційної системи, контейнери ізолюють лише ресурси, від яких залежить ваша програма. Тому вони мають більшу гнучкість ОС, ніж віртуальні машини.

Віртуалізація ОС дозволяє перенести вашу програму на іншу систему, не впливаючи на її розробку та розгортання. Це відповідає потребам багатохмарних та гібридних рішень, зменшуючи ризики прив'язки до постачальника та забезпечуючи плавний перехід з незначними накладними витратами.

## **3. Сумісність з гостями**

Віртуальні машини сумісні майже з усіма операційними системами на хості, тоді як контейнери сумісні лише з версією операційної системи, подібною до версії хоста. Хоча віртуальні машини з'явилися раніше та широко використовувалися протягом багатьох років, контейнери зараз є найкращим вибором для DevOps у багатьох відношеннях.

Нові технології та інновації зазвичай створюються для контейнерних середовищ. Крім того, доступно багато образів контейнерів з відкритим кодом, попередньо зібраних та налаштовуваних. Попередньо зібрані образи для віртуальних машин також доступні, але їх важко налаштувати та забезпечити сумісність.

## **4. Контейнеризація проти віртуалізації: розгортання**

У цьому відношенні контейнеризація також пропонує більшу гнучкість. Існує лише кілька кроків для розгортання програми за допомогою віртуальної машини.

1. Збірка та налаштування віртуальної машини.
2. Встановлення програми на основі цільової ОС.
3. Опублікуйте додаток на платформі на ваш вибір.

Але цей процес тривалий та ресурсомісткий.

Щоб розгорнути контейнеризований застосунок, ви можете отримати образ та розгорнути його на будь-якій сумісній платформі. Це швидкий процес, який не такий ресурсоємний, як віртуальна машина.

## **5. Постійне зберігання**

Контейнери та віртуальні машини відрізняються тим, як їм потрібна інфраструктура зберігання даних та як вона використовується. Це впливає на проектування вашої IT-інфраструктури та допомагає максимізувати цінність середовища застосунків.

Контейнери є тимчасовими, тобто вони запускаються та зупиняються автоматично. Це не стосується віртуальних машин. Однак, обом потрібен доступ до постійного сховища у виробничому середовищі. Без цього доступу контейнер завершує роботу, і дані стають недоступними.

Якщо ви використовуєте контейнеризацію корпоративного рівня, вам потрібно налаштувати постійне середовище зберігання, яке підтримує контейнери.

## **6. Балансування навантаження**

Балансування навантаження віртуальних машин передбачає переміщення запущених віртуальних машин на інші вузли для відмовостійкого кластера. З іншого боку, контейнери

можуть автоматично запускатися або зупинятися на вузлах кластера залежно від змін навантаження.

У цьому випадку змінюється лише доступність, але самі контейнери не переміщуються.

### 7. Нетворкінг

Контейнери використовують ізольований вигляд віртуального мережевого адаптера для легкої віртуалізації, оскільки вони спільно використовують брандмауер хоста. Віртуальні машини використовують повну віртуалізацію з VNA.

Контейнеризація проти віртуалізації: що підходить саме вам?

Як контейнери, так і віртуальні машини (VM) мають свої переваги. Вам слід обрати контейнеризацію, щоб максимізувати кількість програм на мінімальних серверах. Це також правильний вибір для розгортання хмарних програм, упаковки мікросервісів та переміщення масштабованих програм між ІТ-середовищами з однією й тією ж ОС.

Як варіант, віртуальні машини можуть виконувати більше операцій і найкраще підходять для монолітних робочих навантажень, що вимагають повної функціональності ОС. Вибирайте віртуальні машини, якщо ваш застосунок не потребує портативності, і ви хочете розміщувати застарілі застосунки, ізолювати ризиковані середовища розробки та виділити ІТ-ресурси, такі як сервери, сховища та мережі.

Ви також можете використовувати контейнери та віртуальні машини разом для оптимізації ємності та використання сервера. Інтеграція контейнерів у віртуальні машини дозволяє командам DevOps підвищити ефективність фізичного сервера та зменшити кількість збоїв.

Гіпервізор працює в такий спосіб: операційна система хоста емулює апаратне забезпечення, поверх якого вже запускаються гостьові операційні системи. Це означає, що взаємозв'язок між гостьовою й хостовою операційними системами треба «залізній» парадигмі: усе, що «уміє» робити встаткування, повинне бути доступно гостьовий ОС із боку хостової. Навпроти, контейнери – це віртуалізація на рівні операційної системи, а не встаткування, тобто кожна гостьова ОС використовує те ж саме ядро (а в деяких випадках – і інші частини ОС), що й хостова. Це дає контейнерам велику перевагу: вони менше й компактніше гіпервізорних гостьових середовищ, оскільки в них з хостом набагато більше загального.

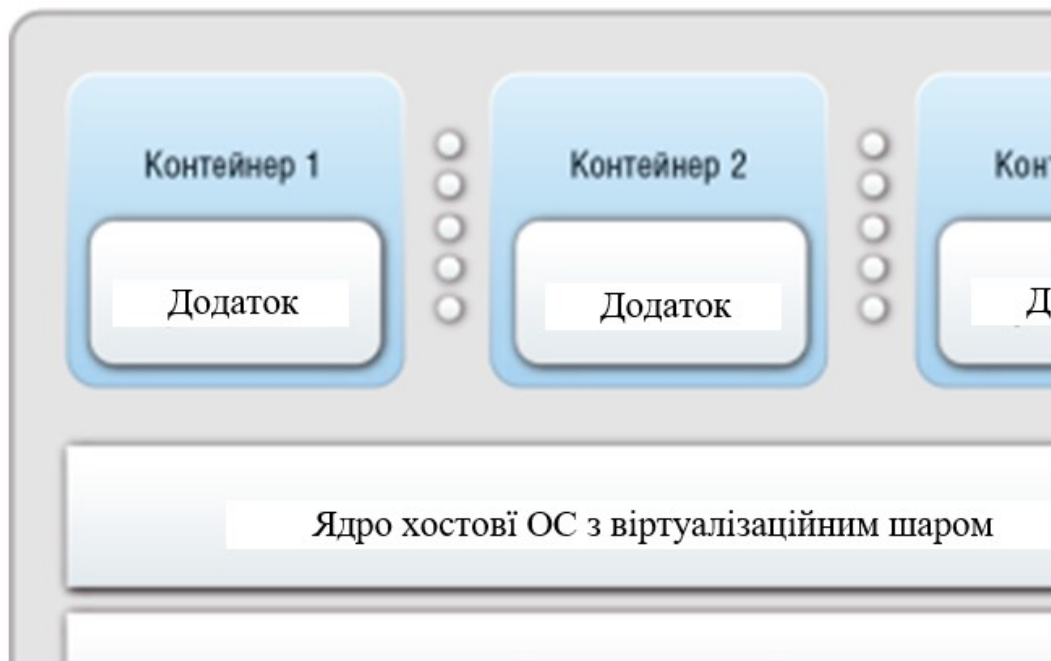


Рисунок 1 – Структурна схема системи

**Висновки.** У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів контейнерної віртуалізації мережевої інфраструктури для хостингу. Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем контейнерної віртуалізації мережевої інфраструктури для хостингу.

– Досліджена система контейнерної віртуалізації мережевої інфраструктури для хостингу.

– На основі отриманих результатів досліджень створена програмна реалізація системи контейнерної віртуалізації мережевої інфраструктури для хостингу.

Розроблені алгоритми дозволяють успішно вирішувати завдання контейнерної віртуалізації мережевої інфраструктури для хостингу. Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

## Список літератури

- Smirnov O., Smirnova T., Anas M. Al-Oraiqat, Drieiev O., Polishchuk L., Sheroz Khan, Yassin M. Y. Hasan, Aladdein M. Amro, Hazim S. AlRawashdeh «Method for Determining Treated Metal Surface Quality Using Computer Vision Technology». *Sensors (Basel, Switzerland) Volume 22, Issue 16*, 6223, 2022.
- Smirnov O., Kuznetsov A., Kryvinska N., Kiiian A., Kuznetsova K. «Full Non-Binary Constant-Weight Codes». *SN Computer Science, Vol 2*, 337, 2021. <https://doi.org/10.1007/s42979-021-00739-w>
- Smirnov O., Kuznetsov A., Zhora V., Onikiychuk A., Pieshkova O. «Hiding Messages in Audio Files Using Direct Spread Spectrum». *11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2021, Cracow, Poland, 22-25 September 2021*. P. 414-418.
- Smirnov O., Kuznetsov A., Lokotkova I., Kuznetsova T., Florov S., Lebid O. «Using Orthogonal Signals to Hide Information in Images». *4 IEEE International Conference on Advanced Information and Communication Technologies (AICT) - 2021, Lviv, Ukraine, September 21-25, 2021*. P. 255-260.
- Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». *International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020*. – P. 247-256.
- Smirnov O., Kuznetsov A., Kiiian A., Kuznetsova T. «Non-binary constant weight coding technique». *CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114*.
- Smirnov O., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». *Journal of theoretical and applied information technology Vol.98. No 21, 2020, P. 3334-3346*.
- Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New technique for data hiding in cover images using adaptively generated pseudorandom sequences». *CEUR Workshop Proceedings Volume 2654, 2020, Pages 1-14*.
- Smirnov O., Kuznetsov A., Onikiychuk A., Makushenko T., Anisimova O., Arischenko A. «Adaptive pseudorandom sequence generation for spread spectrum image steganography». *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18, 2020*. P. 161-165.
- Smirnov O., Kuznetsov A., Kiiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18, 2020*. P. 172-177.
- Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) *Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies, vol 48*. Springer, Cham. 2021. pp 557-587.
- Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». *CEUR Workshop Proceedings Volume 2616, 2020, Pages 366-379*.
- Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», *CEUR Workshop Proceedings Volume 2608, 2020, Pages 633-645*.
- Smirnov, O., Kuznetsov, A., Gorbacheva, L., Babenko, V., «Hiding data in images using a pseudo-random sequence», *CEUR Workshop Proceedings Volume 2608, 2020, Pages 646-660*.
- Zhurakovskiy, B., Tsopa, N., Batrak, Y., Odarchenko, R., Smirnova, T «Comparative analysis of modern formats of lossy audio compression». *Workshop Proceedings, 2020, 2654, стр. 315-327*.
- Smirnov O. Kuznetsov A., Zaichenko Yu., Pastukhov M., Oleshko O., Kuznetsova K., «Formation of Discrete

- Signals with Special Correlation Properties». International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2019; Odessa; Ukraine; 9-13 September 2019. P.22-28.
17. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». International Journal of Computing; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407.
  18. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), Kyiv, Ukraine, 8 – 11 October 2019 . P.517-522.
  19. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». CEUR Workshop Proceedings, Vol 2588, P. 90-106, 2019.
  20. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Averchev, A., Pastukhov, M., Kuznetsova, K., «Formation of Pseudorandom Sequences with Special Correlation Properties», 2019 3rd International Conference on Advanced Information and Communications Technologies, AICT -2019/ Lviv, Ukraine, 2-6 July, 2019, P. 395-399.
  21. Smirnov, O., Kuznetsov, A., Kavun, S., Babenko, B., Nakisko, O., Kuznetsova, K., «Malware Correlation Monitoring in Computer Networks of Promising Smart Grids», 2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS), Kyiv, Ukraine April 17-19, 2019 P. 347-352.
  22. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special Correlation Properties», CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 618-629.
  23. Smirnov A.A., Kuznetsov A.A., Danilenko D.A., Berezovsky A., «The statistical analysis of a network traffic for the intrusion detection and prevention systems», Telecommunications and Radio Engineering. – Volume 74, Issue 1. – Begel House Inc. – 2015. – P. 61-78.
  24. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New Technique for Hiding Data in Cover Images Using Adaptively Generated Pseudorandom Sequences». CEUR Workshop Proceedings Volume 2732, 2020, Pages 214-227.
  25. Т.В. Смірнова, О.М. Дреєв, О.А. Смірнов «Хмарна інформаційна система оцінювання шорсткості з використанням дискретного частотного аналізу макروفотografій». IV міжнародна науково-практична конференція «Інформаційна безпека та комп'ютерні технології», м. Кропивницький. 15-16 квітня 2021р. – Кропивницький: ЦНТУ. – 2021. – С. 30.
  26. О.А. Смірнов, П.С. Усік, «дослідження перспектив використання технологічних рішень в мережах 5g» у Кібербезпека та інформаційні технології: монографія. – Х. : ТОВ «ДІСА ПЛЮС», 2020.С. 122-135.
  27. О.А.Смірнов, Т.В.Смірнова, Л.І. Поліщук, К.О. Буравченко, А.О.Макевнін, «Дослідження хмарних технологій як сервісів», Кібербезпека: освіта, наука, техніка. № 3(7). С. 43-62. 2020.
  28. Смірнов О.А., Дреєва Г.М., Дреєв О.М., Смірнова Т.В. «Фрактальний аналіз генератора самоподібного трафіку на основі ланцюга Маркова». Центральноукраїнський науковий вісник. Технічні науки. № 2(33). с. 161-172, 2019.
  29. О. Смірнов, Є. Деменко, О. Онікійчук, А. Арищенко, Л. Горбачова, «Формування псевдовипадкових послідовностей для приховування даних в зображеннях» Комп'ютерні науки та кібербезпека. № 4. С. 30-37. 2019.
  30. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В. Поліщук Л.І. Проектування комп'ютерних систем та мереж. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2019. – 264 с.
  31. Smirnov, O., Kuznetsov, A., Kuznetsova., K. Synthesis of Discrete Signals with Improved Correlation Properties. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).