Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Метод виявлення вторгнень у розподілених інформаційних системах

(тема)

Виконав:

студент II курсу, групи СПм-21-1

Кізлевич Я.О.
(прізвище, ініціали)

Спеціальність

123 «Комп’ютерна інженерія»
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма

Системне програмування
(повна назва освітньої програми)

Керівник: доц. Мартовицький В.О.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ _____ Коваленко А.А.
(підпис) (прізвище, ініціали)

2022 р.

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

Спеціальність 123 «Комп’ютерна інженерія»
(код і повна назва)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне програмування
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
“____” _____ 20___ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Кізлевичу Яну Олександровичу
(прізвище, ім’я, по батькові)

1. Тема роботи Метод виявлення вторгнень у розподілених інформаційних системах

затверджена наказом по університету від “ 07 ” листопада 2022 р. № 1454 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 13 грудня 2022 р.

3. Вхідні дані до роботи Навчальна вибірка

4. Перелік питань, що потрібно опрацювати у роботі

1. Аналіз сучасних систем виявлення вторгнень.
2. Аналіз методів виявлення вторгнень
3. Розробка метода виявлення вторгнень у розподілених інформаційних системах
4. Тестування та аналіз роботи розробленого методу

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) 14 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Огляд методів виявлення вторгнень	08.11.22–11.11.22	
2	Вибір та обґрунтування методики дослідження	12.11.22–17.11.22	
3	Аналіз систем виявлення вторгнень	18.11.22–21.11.22	
4	Вибір методів машинного навчання моделі	22.11.22–28.11.22	
5	Проведення експериментальних досліджень	29.11.22–02.12.22	
6	Оформлення матеріалів кваліфікаційної роботи	03.12.22–06.12.22	
7	Подання кваліфікаційної роботи керівникові	07.12.22–08.12.22	
	та її попередній захист		
8	Подання кваліфікаційної роботи на	09.12.22–12.12.22	
	рецензування		

Дата видачі завдання 07 листопада 2022 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Мартовицький В. О.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 64 с., 12 рис., 9 табл., 1 дод., 24 джерел.

ІНФОРМАЦІЙНА СИСТЕМА, АНОМАЛІЇ, КІБЕРБЕЗПЕКА, ПРОТОКОЛ, СЕРВЕР, ШЛЮЗ, FIREWALL.

Метою кваліфікаційної роботи є покращення показників виявлення аномалій функціонування РІС в умовах кібернетичних впливів зовнішнього та внутрішнього середовища шляхом побудови методу на основі технологій інтелектуального аналізу даних.

Для досягнення поставленої мети вирішуються наступні задачі:

– проаналізувати підходи щодо забезпечення безпеки розподілених інформаційних систем;

– розглянути особливості архітектури розподілених інформаційних систем і виділити базові компоненти систем;

– розробити метод виявлення аномалій системи з використанням методів інтелектуального аналізу для класифікації стану функціонування комп’ютерних систем.

ABSTRACT

Master’s thesis: 64 pages, 12 figures, 9 tables, 1 appendices, 24 sources.

INFORMATION SYSTEM, ANOMALIES, CYBER SECURITY, PROTOCOL, SERVER, GATEWAY, FIREWALL.

The major goal of this thesis is to improve the indicators of the detection of anomalies in the functioning of RIS in the conditions of cybernetic influences of the external and internal environment by building a method based on the technologies of intelligent data analysis.

To achieve the goal, the following tasks are solved:

– analyze approaches to ensuring the security of distributed information systems;

– consider the features of the architecture of distributed information systems and highlight the basic components of the systems;

– to develop a method of detecting system anomalies using methods of intellectual analysis to classify the state of functioning of computer systems.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

EAI – інтеграція корпоративних додатків (англ., Enterprise Application Integration)

IDS – система виявлення вторгнень (англ., Intrusion Detection System)

IPS – система попередження вторгнень (англ., Intrusion Prevention System)

IT – інформаційні технології (англ., Information Technology)

ITRC – Міждержавна технологічна та регуляторна рада (англ., Interstate Technology and Regulatory Council)

MFA – багатофакторна автентифікація (англ., Multi-Factor Authentication)

REST – передача репрезентативного стану (англ., Representational State Transfer)

SLA – угода про рівень послуг (англ., Service-Level Agreement)

SOA – сервіс-орієнтована архітектура (англ., Service-Oriented Architecture)

SSO – технологія єдиного входу (англ., Single Sign-On)

ВСТУП

Розуміння стану інфраструктури та систем важливо для стабільної роботи сервісів. Інформація про працездатність і продуктивності розгортання не тільки допомагає команді вчасно реагувати на проблеми, але і дає їм можливість впевнено вносити всі необхідні зміни. Один з кращих способів отримати цю інформацію є надійна система моніторингу, яка збирає метрики системи, візуалізує дані і попереджає операторів, про кібернетичні впливи на комп’ютерну систему.

У системах моніторингу мережної інфраструктури відбуваються радикальні зміни, викликані загостренням конкуренції на ринку, зростанням вимог до якості забезпечення безпеки, технічним переозброєнням мереж зв'язку, зміною характеру розподілу трафіка. Все це призводить до необхідності здійснення контролю великої кількості параметрів функціонування мереж різних технологій.

Система моніторингу не тільки змінює уявлення про систему експлуатації, переходячи від збору даних параметрів окремих станцій до параметрів експлуатації всієї мережі, а також автоматизує безліч рутинних процесів зі збору та обробки параметрів розподіленої комп’ютерної системи.

Аналіз цієї інформації дає можливість виявлення різноманітних випадків загроз та порушень, таких як:

- несанкціоноване підключення до мережі, пропущене класичними засобами захисту периметра (IPS / IDS);
- поширення вірусів і шпигунського програмного забезпечення, не виявлених штатними антивірусними засобами;
- неправильні дії при користуванні ресурсами розподілених комп’ютерних систем;
- підключення в мережу нових пристроїв і їх поведінка;
- помилки в роботі обладнання;

- виникнення в мережі «вузьких» місць і інші можливі порушення.

Архітектура системи моніторингу параметрів отриманих від датчиків, характеризуються не тільки їх цільовими функціями, але і функціональними можливостями, що забезпечують реалізацію цільових функцій, ієрархією та рівень паралелізму вирішення завдань, однорідністю або різнорідністю модульної структури, організацією збору інформації в режимі реального часу, обробки даних і мережного обміну інформацією з абонентами.

При цьому повинні забезпечуватися:

- невтручання в роботу мережного обладнання;
- постійний збір статистичної інформації, який дозволяє створювати повномасштабні бази даних, необхідні для проведення аналізу параметрів мережі в масштабі реального часу;
- забезпечення високої швидкості обробки запитів на надання потрібних інформаційних ресурсів і сервісів;
- виконання збору, обробки, зберігання повної інформації про стан всіх компонентів телекомунікаційної і інформаційної інфраструктури мережі в реальному часі незалежно від архітектури мережі, типу комутатора і постачальника;
- створення єдиного стандартизованого інформаційного центру зберігання даних про стан систем і мережі.

З огляду на великий обсяг подій, що супроводжують процес діагностичного моніторингу, різноманіття типів подій і пристроїв у відкритій системі, що діагностується, і необхідність функціонування в режимі реального часу з урахуванням високої мінливості зовнішнього середовища, завдання побудови діагностичного моніторингу мережі слід віднести до проблематики обробки великих даних. Рішення даної проблеми пов'язане з реалізацією нових парадигм розробки програмних систем, підтримуючих можливість розподіленої взаємодії автономних активних пристроїв в процесі вирішення конкретного оперативного завдання.

1 АНАЛІЗ СУЧАСНОГО СТАНУ ПИТАННЯ ВИЯВЛЕННЯ АНОМАЛІЙ В ІНФОРМАЦІЙНИХ СИСТЕМАХ

1.1 Аналіз сучасного стану інформаційних систем

У 21 столітті інформаційні системи вирішують завдання не тільки науково-технічної діяльності, а й використовуються у всіх областях діяльності людини. Ці технології розвиваються стрімко і мають великий потенціал.

Інформаційні системи ніколи не існую самі по собі. Вони завжди пов'язані з якоюсь діяльністю людини (організації): розрахунком при моделюванні геофізичних процесів для вирішення завдань розвідки і видобутку корисних копалин, управлінням безпекою руху поїздів, обліком пацієнтів лікарні для аналізу їх симптомів та допомоги у встановленні діагнозу, розрахунком заробітної плати, обліком нерухомості, пошуком веб-сторінок, реконструкцією археологічних об'єктів та ін.

Діяльність, пов'язана безпосередньо з інформаційними системами (і тільки з ними), рідко буває основною (якщо тільки організація не зайнята виключно розробкою і / або супроводом ІС). Інформаційна система завжди тільки обслуговує основну діяльність організації або людини.

Інформаційна система — організаційно-технічна система, в якій реалізується технологія обробки інформації з використанням технічних і програмних засобів [1].

У будь-якій інформаційній системі організовуються певні процеси, щоб[2]:

- виявити інформаційні потреби;
- здійснити відбір джерел інформації;
- здійснити збір інформації;
- виконати дії з обробки інформації, оцінці її повноти і значущості та за поданням її в зручному вигляді;

- вивести інформацію для надання споживачам або передачі в іншу систему;

- організувати використання інформації для оцінки тенденцій, розробки прогнозів, оцінки альтернатив рішень і дій, вироблення стратегії;

- організувати зворотний зв'язок – за результатами обробки даних здійснити корекцію взаємодії із зовнішнім середовищем.

Всі ці дії здійснюються за допомогою тих чи інших інформаційних технологій в рамках інформаційної системи організації.

На рисунку 1.1 представлено загальні функції інформаційної системи, які спільні для будь-якої системи.

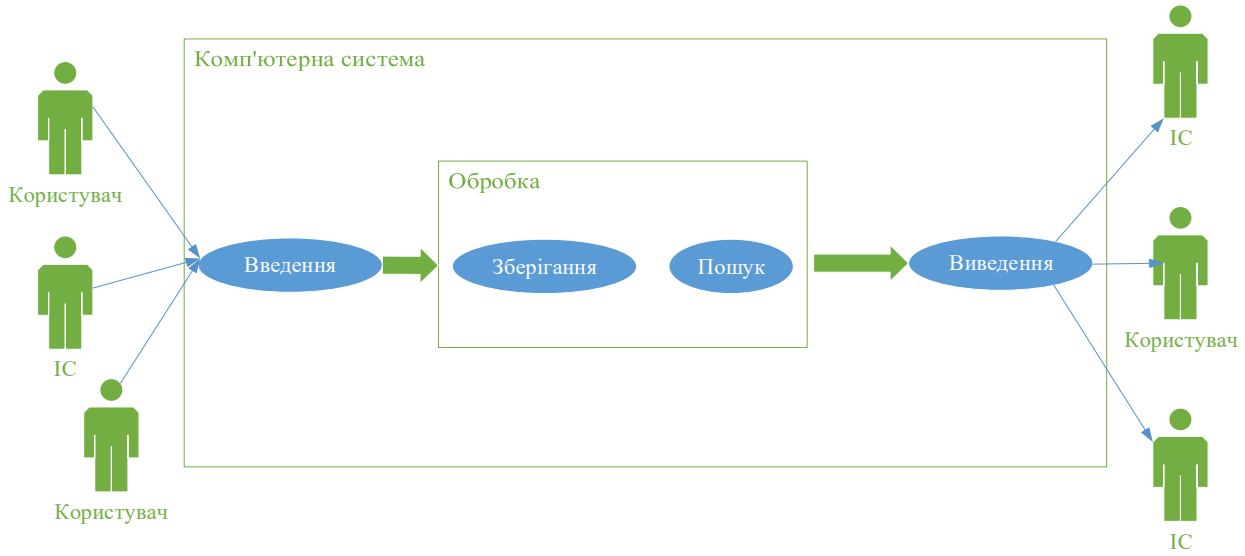

Рисунок 1.1 – Загальні функції інформаційної системи

Кожна функція інформаційної системи може виконуватися окремим компонентом ІС. Такий компонент називається підсистемою або модулем (в залежності від оціненої складності або розміру компонента). У невеликих ІС підсистема може реалізувати кілька функцій; у великих і складних ІС їх функції деталізуються (найпростіший приклад - поділ функцій зберігання і обробки інформації). Кожна така детальна функція може реалізовуватися своєї підсистемою; підсистеми можуть реалізовувати кілька різних детальних

функцій.

У кожній з таких підсистем можуть загрожувати випадкові або навмисні збої (аномалії). Під загрозою будемо розуміти потенційно можливі дії на підсистему, які прямо або побічно можуть завдати шкоди користувачеві. Безпосередню реалізацію загрози називають атакою. Має сенс розрізняти ненавмисні і навмисні загрози.

Ненавмисні загрози пов'язані з:

- помилками обладнання або програмного забезпечення: збої процесора, живлення, проблемами при зчитувані з диска, помилки в комунікаціях, помилки в програмах;
- помилками людини: некоректне введення, неправильне монтування дисків, запуск невірних програм, втрата дисків, пересилання даних за невірною адресою;
- форс-мажорними обставинами.

Навмисні загрози, на відміну від випадкових, мають на меті нанесення шкоди користувачам інформаційних систем і, в свою чергу, поділяються на активні і пасивні. Пасивна загроза - несанкціонований доступ до інформації без зміни стану системи, активна - пов'язана зі спробами перехоплення і зміни інформації.

Не існує загальноприйнятої класифікації загроз безпеки. Один з варіантів класифікації може бути виконаний за такими ознаками:

- по цели реализации;
- по принципу воздействия на систему;
- по характеру воздействия на систему;
- по причине появления используемой ошибки защиты;
- по способу воздействия атаки на объект;
- по объекту атаки;
- по используемым средствам атаки;
- по состоянию объекта атаки.

До найбільш поширених загроз безпеки відносять:

- несанкціонований доступ - найбільш поширений вид комп'ютерних порушень. Він полягає в отриманні користувачем доступу до ресурсу, на який у нього немає дозволу відповідно до прийнятої в організації політики безпеки;

- відмова в обслуговувані. Являє собою навмисне блокування легального доступу до інформації та інших ресурсів;

- незаконне використання привілеїв. Зловмисники, які застосовують даний спосіб атаки, зазвичай використовують штатний програмне забезпечення, яке функціонує в позаштатному режимі. Незаконне захоплення привілеїв можливе або при наявності помилок в самій системі, або в разі недбалості при керуванні системою. Суворе дотримання правил управління системою захисту, дотримання принципу мінімуму привілеїв дозволяє уникнути таких порушень;

- «приховані канали». Являють собою шляху передачі інформації між процесами системи, що порушують системну політику безпеки. У середовищі з поділом доступу до інформації користувач може не отримати дозвіл на обробку даних, які його цікавлять, однак може знайти для цього обхідні шляхи. «Приховані канали» можуть бути реалізовані різними шляхами, зокрема за допомогою програмних закладок;

- «маскарад». Під «маскарадом» розуміється виконання будь-яких дій одним користувачем від імені іншого користувача. Такі дії іншому користувачеві можуть бути дозволені. Порушення полягає в привласненні прав і привілеїв;

- «збір логів». Після закінчення роботи інформація, яка оброблялася не завжди повністю видаляється з пам'яті ПК. Дані зберігаються на носії до перезапису або знищення; при виконанні цих дій на звільненому просторі диска знаходяться їх залишки. При спотворенні заголовка файлу їх прочитати важко, але все ж можливо за допомогою спеціальних програм і обладнання. Такий процес прийнято називати «збором сміття». Він може привести до витоку важливої інформації;

- «люки». Являють собою приховану, не задокументовану точку входу

в програмний модуль. «Люки» відносяться до категорії загроз, що виникають внаслідок помилок реалізації будь-якого проекту (системи в цілому, комплексу програм та інше). Тому в більшості випадків виявлення «люків» - результат випадкового пошуку;

- шкідливі програми. Останнім часом почастішали випадки впливу на обчислювальну систему спеціально створеними програмами. Для позначення всіх програм такого роду був запропонований термін «шкідливі програми». Ці програми прямо або побічно дезорганізують процес обробки інформації або сприяють витоку або спотворення інформації.

До найпоширеніших видів подібних програм відносяться:

- «вірус» - це програма, яка здатна заражати інші програми, модифікуючи їх так, щоб вони включали в себе копію вірусу;

- «троянський кінь» - програма, яка містить прихований або явний програмний код, при виконанні якого порушується функціонування системи безпеки. «Троянські коні» здатні розкрити, змінити або знищити дані або файли. Їх вбудовують в програми широкого користування, наприклад, в програми обслуговування мережі, електронної пошти;

- «черв'як» - програма, яка розповсюджується в системах і мережах по лініях зв'язку. Такі програми подібні вірусам: заражають інші програми, а відрізняються від вірусів тим, що не здатні самовідтворюватися;

- «жадібна» програма- програма, яка захоплює (монополізує) окремі ресурси обчислювальної системи, не даючи іншим програмам можливості їх використовувати;

- «бактерія» - програма, яка робить копії самої себе і стає паразитом, перевантажуючи пам'ять ПК і процесор;

- «логічна бомба» - програма, яка веде до пошкодження файлів або комп'ютерів (від спотворення даних - до повного знищення даних). «Логічний бомбу» вставляють, як правило, під час розробки програми, а спрацьовує вона при виконанні деякої умови (час, дата, введення кодового слова);

- «Adware» - точка входу в програму, завдяки якій відкривається доступ

до деяких системних функцій. Виявляється шляхом аналізу роботи програми.

Перераховані атаки часто використовуються спільно для реалізації комплексних атак. Так, наприклад, троянська програма може використовуватися для збору інформації про користувачів на віддаленому комп'ютері і пересилання її зловмисникові, після чого останній може здійснити атаку методом «маскарад».

Забезпечення інформаційних систем поділяється на: інформаційне, технічне, математичне і програмне, методичне, лінгвістичне, правове і організаційне (рисунок 1.2).

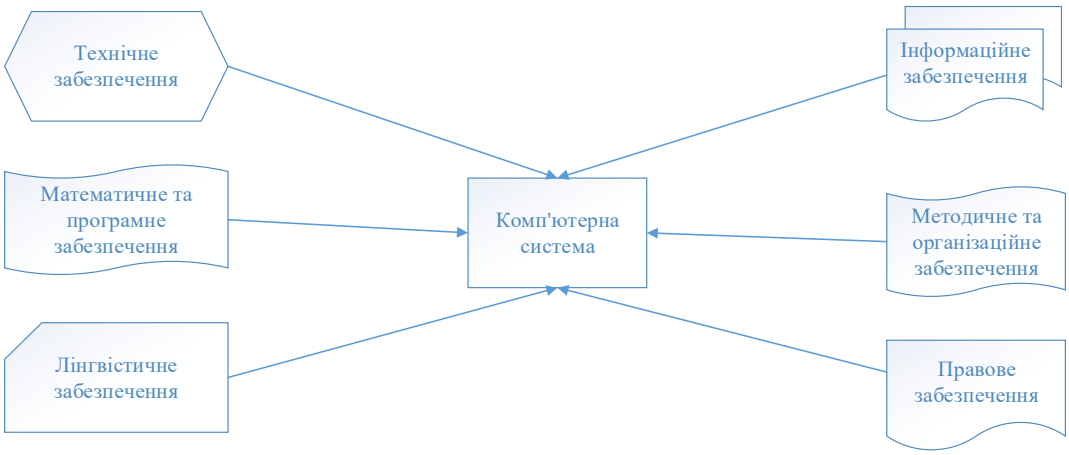

Рисунок 1.2 – Забезпечення комп’ютерних систем

Інформаційне забезпечення — комплексне поняття, що охоплює сукупність даних, організацію їх введення, обробки, збереження та накопичення, пошуку, а також поширення в межах компетенції зацікавлених осіб в зручному для них вигляді. Складовими інформаційного забезпечення виділено інформаційні технології, інформаційні ресурси, технічні засоби та програмне забезпечення. [3]

Технічне забезпечення – це комплекс різних видів техніки: обчислювальна техніка, периферійні пристрої, засоби автоматичного зчитування даних, офісні обладнання, комунікаційне обладнання, засоби передачі та обміну даними, комунікаційне обладнання, мережеве обладнання,

засоби мультимедіа тощо [4].

Математичне забезпечення - сукупність математичних методів, моделей, алгоритмів обробки інформації, використовуваних при вирішенні функціональних і проектних завдань в ІС.

Під програмним забезпеченням інформаційних комп'ютерних технологій розуміють сукупність програмних і документальних засобів для створення та експлуатації систем обробки даних засобами обчислювальної техніки [2].

Методичне та організаційне забезпечення - комплекс методів, засобів і документів, що регламентують взаємодію персоналу ІС з програмно-технічними засобами і між собою в процесі розробки і експлуатації ІС.

Лінгвістичне забезпечення - сукупність мов спілкування персоналу ІС і користувачів з програмно-технічним та інформаційним забезпеченням, а також перелік термінів, які використовуються в ІС.

Правове забезпечення - правові норми, які використовуються для дотримання законності (закони, укази, постанови державних органів влади, накази та інструкції вищестоящих органів і керівників організації).

До загрози підсистеми інформаційного забезпечення відносяться:

- порушення адресності та своєчасності інформаційного обміну, протизаконний збір і використання інформації;
- несанкціонований доступ до інформаційних ресурсів;
- маніпулювання інформацією (дезінформація, приховування або перекручення інформації);
- незаконне копіювання даних в інформаційних системах;
- порушення технології обробки інформації.

До загрози підсистеми програмно-математичного забезпечення відносяться:

- впровадження комп'ютерних вірусів;
- встановлення програмних і апаратних закладних пристроїв;
- знищення або модифікацію даних в автоматизованих інформаційних

системах.

- До загрози підсистеми технічного забезпечення відносяться:
- перехоплення інформації в технічних каналах її можливого витоку;
- впровадження електронних пристроїв перехоплення інформації в технічні засоби і приміщення;
- перехоплення, дешифрування і нав'язування неправдивої інформації в мережах передачі даних і лініях зв'язку;
- вплив на парольно-ключові системи;
- радіоелектронне придушення ліній зв'язку та систем управління.
- До загрози підсистем організаційно-правового забезпечення відносяться:
- невиконання вимог законодавства та затримки в прийнятті необхідних нормативно-правових положень в інформаційній сфері;
- неправомірне обмеження доступу до документів, що містять важливу для громадян і організацій інформацію.
- До загрози підсистеми лінгвістичного забезпечення відносяться:
- некоректна обробка вхідних даних, що використовуються в SQL-запитах;
- впровадження шкідливого коду.

1.2 Аналіз розподілених інформаційних системи

Сучасні технології вимагають високих швидкостей обробки інформації, зручних форм її зберігання і передачі. Необхідно також мати динамічні способи звернення до інформації, способи пошуку даних в задані тимчасові інтервали, щоб реалізовувати складну математичну і логічну обробку даних.

Управління великими підприємствами, економікою на рівні країни вимагають участі в цьому процесі досить великих колективів. Такі колективи можуть розташовуватися в різних районах міста, в різних регіонах країни і навіть в різних країнах. Для вирішення завдань управління, що забезпечують

реалізацію економічної стратегії, стають важливими і актуальними швидкість і зручність обміну інформацією, а також можливість тісної взаємодії всіх що у процесі вироблення управлінських рішень.

В епоху централізованого використання ЕОМ з пакетної обробкою інформації користувачі обчислювальної техніки вважали за краще купувати комп'ютери, на яких можна було б вирішувати майже всі класи їх завдань. Однак складність вирішуваних завдань обернено пропорційна їх кількості, і це призводило до неефективного використання обчислювальної потужності ЕОМ при значних матеріальних витратах. Не можна не враховувати і той факт, що доступ до ресурсів комп'ютерів був утруднений через існуючу політики централізації обчислювальних засобів в одному місці.

Принцип централізованої обробки даних не відповідав високим вимогам до надійності процесу обробки, утруднював розвиток систем і не міг забезпечити необхідні тимчасові параметри при діалогової обробці даних в багато користувацькому режимі. Короткочасний вихід з ладу центральної ЕОМ приводив до фатальних наслідків для системи в цілому.

З появою персональних комп'ютерів з’явився новий підхід до організації систем обробки даних та до створення нових інформаційних технологій. Виникло логічно обгрунтована вимога переходу від використання окремих ЕОМ в системах централізованої обробки даних до розподіленої обробки даних.

Слід відзначити, що ключовим компонентом розподіленої системи є відповідний шар програмного забезпечення (ПЗ), що надає можливість скоординованої роботи визначеної кількості гетерогенних ресурсів - фізичних обчислювачів та програмних додатків, які функціонують під керуванням операційних систем (ОС) (рисунок 1.3). Цей шар базується на стандартах представлення та обміну (representation and exchange) інформацією. Особливостями ресурсів є можливість незалежного використання, скоординована робота (collaboration) з метою вирішення обчислювальної задачі, доступність до кінцевого користувача. Об’єднання ресурсів повинно

здійснюватись у вигляді єдиної, несуперечливої (coherent) системи. Важливою складовою вищеозначеного шару ПЗ РІС являється набір методів, підходів, засобів розподілення та управління вирішенням обчислювальних завдань (tasks), множина яких складає обчислювальну задачу (computing problem). Безумовно ПЗ РІС можливо розглядати на різних рівнях, виділити додаткові шари, класи компонентів залежно від їх специфіки. Але основу такого ПЗ складає ряд сервісів, які формують інтерфейс взаємодії компонентів різних програмних додатків та вирішують проблеми, пов’язані з відмінностями апаратних та програмних платформ.

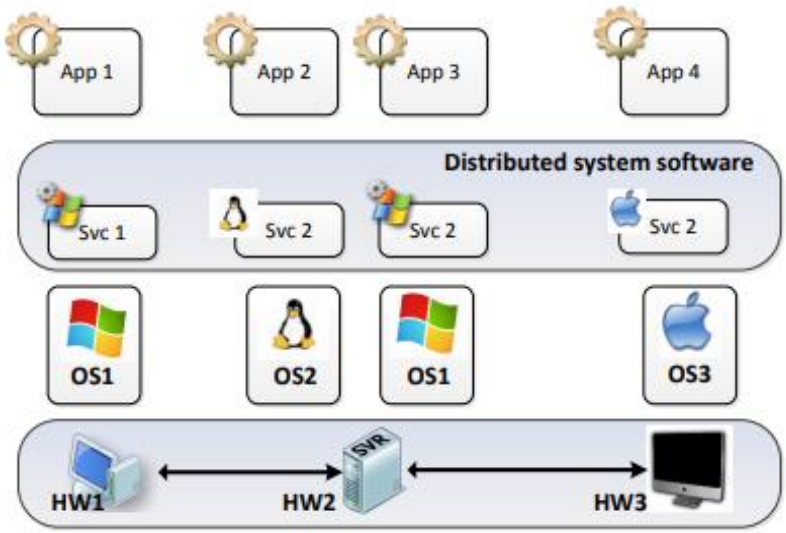

Рисунок 1.3 – Модель розподіленої комп’ютерної системи

Розподілена обробка даних (Distributed Data Processing, DDP) – це така обробка, при якій все або деякі функції обробки, зберігання і введення-виведення даних разом з функціями управління розосереджені на декількох станціях. Метою такого розосередження є ефективна обробка інформації з точки зору функціональних і економічних можливостей або географічного розташування систем. Розподілена обробка даних передбачає розподіл комп’ютерів, обчислювальних завдань і самих даних[5].

РІС будуються на основі мережних технологій і являють собою

обчислювальні мережі. Комутаційна підсистема яких включає:

- комутаційні модулі;
- канали зв'язку;
- концентратори;

В РІС всі потенційно навмисні загрози безпеки інформації ділять на дві групи: пасивні та активні.

До пасивних відносяться загрози, метою реалізації яких є отримання інформації про систему шляхом прослуховування каналів зв'язку.

Активні загрози передбачають вплив на передані повідомлення в мережі і несанкціоновану передачу фальсифікованих повідомлень з метою впливу на інформаційні ресурси об'єктів РІС та дестабілізацію функціонування системи. Можливо також безпосередній вплив на комунікаційну підсистему з метою пошкодження апаратних засобів передачі інформації.

Передані в РІС повідомлення можуть несанкціоновано модифікуватися або знищуватися. Зловмисник може розмножувати перехоплені повідомлення, порушувати їх черговість проходження, змінювати маршрут доставки, підміняти повідомлення, можуть бути спроби несанкціонованого доступу до інформаційних ресурсів віддаленого об'єкта РІС, здійснення несанкціонованого зміни програмної структури РІС шляхом впровадження шкідницьких програм.

У РІС більшість загроз пов'язані з передачею інформації по каналах зв'язку, з територіальної роз'єднаністю об'єктів системи.

Тому, в РІС, поряд із заходами, що застосовуються для забезпечення безпеки інформації в окремих об’єктах системи, реалізується ряд механізмів для захисту інформації при передачі її по каналах зв'язку.

В основі розподілених обчислень лежать дві основні ідеї: багато організаційно і фізично розподілених користувачів, що одночасно працюють із загальними даними - загальною базою даних (користувачі з різними іменами, які можуть розташовуватися на різних обчислювальних установках, з різними повноваженнями і завданнями); логічно і фізично розподілені дані,

що становлять і утворюють тим не менш, загальну базу даних (окремі таблиці, записи і навіть поля можуть розташовуватися на різних обчислювальних установках або входити в різні локальні бази даних).

Ключовими характеристиками розподілених систем являються: прозорість, відкритість, паралельність виконання завдань, масштабованість, стійкість до помилок [6].

Прозорість (Transparency).

Дана характеристика дозволяє приховувати деталі реалізації, структури й функціонування системи. Щодо структурної складової системи - приховуються особливості представлення даних, наявність ряду різнорідних програмних і апаратних ресурсів, особливості їх конкурентного використання. Стосовно процесів − приховуються наявність паралельної обробки (concurrency), міграції (migration) або переміщення (reallocation) ресурсів. Міграція на відмінність від переміщення означає «живе» переміщення ресурсу або обчислювального процесу на інший вузол РІС без зупинки процесу користування (наприклад, live migration в системі Microsoft Hyper-V користувальницьких віртуальних машин з одного вузла кластера на інший для збалансування потужності). Важливою функцією розподіленої системи є приховування деталей та навіть факту розподілення обчислень між багатьма апаратними та програмними ресурсами, деталей координації їх роботи. Основні види прозорості, які забезпечуються розподіленими системами наведені у таблиці 1.1.

Таблиця 1.1 – Типи прозорості

Тип	Опис
Доступ (Access)	Приховує різноманіття в представленні даних та в питаннях доступу до ресурсів
Розміщення (Location)	Дозволяє забезпечити доступ до локальної та віддаленої інформації в уніфікованому вигляді незалежно від місця та часу взаємодії

Продовження таблиці 1.1

Помилок(Failure)	Дозволяє автоматично виявити і виправити помилки, створюючи при цьому у користувача враження безпомилкової роботи системи
Реплікації (Replication)	Дозволяє невидимо дублювати програмне забезпечення і дані на ряді машин з метою підвищення продуктивності
Міграції (Migration)	Приховує переміщення ресурсу на інше місце

Інша важлива характеристика розподілених систем - це відкритість.

Відкрита розподілена система {open distributed system) - це система, яка пропонує служби, виклик яких вимагає стандартний синтаксис і семантику.

У розподілених системах служби зазвичай визначаються через інтерфейси (interfaces), які часто описуються за допомогою мови опису інтерфейсів (Interface Definition Language, IDL), Опис інтерфейсу на IDL виключно стосується синтаксису служб. Іншими словами, вона точно відображає імена доступних функцій, типи параметрів, значень, що повертаються, виняткові ситуації, які можуть бути порушені службою та інше. Найбільш складно точно описати те, що робить ця служба, тобто семантику інтерфейсів. На практиці подібні специфікації задаються неформально, за допомогою природної мови.

Важливо відзначити, що специфікація не визначає зовнішній вигляд реалізації, вона повинна бути нейтральною. Самодостатність і нейтральність необхідні для забезпечення універсальності і здатності до взаємодії [7]. Здатність до взаємодії (interoperability) характеризує, наскільки дві реалізації систем або компонентів від різних виробників в змозі спільно працювати, покладаючись тільки на те, що служби кожної з них відповідають загальному стандарту (інтерфейсу).

Переносимість (poitability) характеризує те, наскільки додаток, розроблений для розподіленої системи А, може без змін виконуватися в

розподіленій системі В, реалізуючи ті ж інтерфейси , що і в системі А.

Наступна важлива характеристика відкритих розподілених систем – це гнучкість.

Гнучкість – простота конфігурування системи, що складається з різних компонентів різних виробників (можливість переносу системи з одних операційних систем на інші, можливо навіть в іншу файлову систему). Не повинні викликати ускладнень додавання до системи нових компонентів або заміни існуючих, при цьому інші компоненти, з якими не проводилося ніяких дій, повинні залишатися незмінними. Для забезпечення масштабованості необхідно використовувати децентралізовані дані й алгоритми, служби [8].

Масштабованість - це одна з найбільш важливих завдань при проектуванні розподілених систем. Масштабованість системи може вимірюватися за трьома різними показниками [9]:

- по-перше, система може бути масштабованою по відношенню до її розміру, що означає легкість підключення до неї додаткових користувачів і ресурсів.

- по-друге, система може масштабуватися географічно, тобто користувачі і ресурси можуть бути рознесені в просторі.

- по-третє система може бути масштабованої в адміністративному сенсі, тобто бути проста в управлінні при роботі в множині адміністративно незалежних організацій.

На жаль, система, що володіє масштабованістю по одному або декільком з цих параметрів, при масштабуванні часто втрачає продуктивності.

Незважаючи на всі переваги розподілених систем в порівнянні з традиційними централізованими системами, РС мають і ряд істотних недоліків.

Основними загрозами розподілених систем є:

- перехоплення;
- переривання;
- модифікація;

- підробка.

Перехопленням називається така ситуація, коли неавторизований агент отримує доступ до служб або даними. Типовий приклад перехоплення - коли зв'язок між двома агентами підслуховує хтось третій.

Прикладом переривання може бути пошкодження або втрата файлу. Зазвичай переривання пов'язують з такою ситуацією, коли служби або дані стають недоступними, знищуються, їх неможливо використовувати та інше. У цьому сенсі атаки типу «відмова в обслуговуванні», при яких хтось навмісно намагається зробити певну службу недоступною для інших, - це загроза захисту, що класифікується як переривання.

Модифікації включають в себе неавторизовані зміни даних або фальсифікацію служб з тим, щоб вони не відповідали своєму оригінальному призначенню. Приклади модифікації включають перехоплення повідомлень з наступною зміною переданих даних, фальсифікацію входів в бази даних і зміна програм з тим, щоб таємно відстежувати діяльність користувачів.

Підробці відповідає ситуація, коли створюються додаткові дані або здійснюється діяльність, неможлива в нормальних умовах. Так, наприклад, зловмисник може спробувати додати записи в файл паролів або базу даних. Крім того, іноді вдається увійти в систему, відтворивши відправку раніше посланого повідомлення.

2 АНАЛІЗ ПІДХОДІВ ЩОДО ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ РОЗПОДІЛЕНИХ КОМП’ЮТЕРНИХ СИСТЕМ

2.1 Аналіз підходів щодо забезпечення безпеки розподілених комп’ютерних систем

Відомо, що в зв'язку з просторовою розподіленістю, різноманітністю режимів функціонування і структурної складністю сучасних комп'ютерних мереж, програмне забезпечення і інформація, що обробляється виявляються дуже вразливими. У числі факторів, що впливають на безпеку розподілених комп’ютерних систем, зазвичай виділяють наступні фактори: велика кількість суб'єктів, що мають доступ до системи, концентрація в базі даних великих обсягів інформації необхідних користувачеві в різних віддалених вузлах мережі, різноманітність моделей та видів технічних і програмних засобів, варіантів доступу, наявність протяжних ліній зв'язку, необхідність використання при передачі даних проміжних вузлів, спільне використання ресурсів багатьма користувачами, необхідність узгодженого спільного функціонування декількох вузлів мережі в процесі розподіленої обробки інформації, розмитість меж мережі - один і той же вузол може бути доступний користувачам різних мереж, безліч можливих точок атаки особливо при наявності доступу по комутованих лініях зв'язку, складність контролю доступу до системі з віддалених точок мережі, велика кількість комбінацій різних програмно-апаратних засобів і режимів їх роботи.

Тільки за рахунок несанкціонованого доступу можна десятками різних способів знизити працездатність мережі. І зловмисники успішно використовують ці можливості - число злочинів в сфері інформаційних технологій щорічно збільшується.

Питання кіберзлочинності в сфері інформаційних технологій досліджували такі вчені, як: Б. Головін, В. Голіна, А. Голуб та інші.

Щорічно кількість виявлених кіберзлочинів збільшується в середньому на 2500. У 2017 році кіберполіція супроводжувала близько 7 тис. кримінальних проваджень, з них 4,5 тис. - виключно кіберзлочини. За 11 місяців 2017 року кіберполіція направила до суду обвинувальні акти за 726 особами[7].

Під інформаційною безпекою розуміється захист інформації і підтримка цілісності її інфраструктури за допомогою сукупності програмних, апаратно-програмних засобів і методів, а також організаційних заходів, з метою запобігти завданню шкоди власникам цієї інформації або підтримуючої її інфраструктури.

При побудові системи захисту повинен враховуватися комплексний підхід у забезпеченні безпеки інформації. Мається на увазі використання захисних механізмів на всіх етапах життєвого циклу системи, від її проектування і до виведення в експлуатацію, і спільне рішення цілого спектру питань, починаючи від фізичного захисту об'єктів РКС, із застосуванням інтелектуальної системи контролю доступу, і закінчуючи питаннями підтримки нормального функціонування РКС в критичних ситуаціях.

Проектування системи безпеки інформації (СБІ) здійснюється спільно з проектуванням самої комп’ютерної системи. При внесенні будь-яких змін в структуру РКС, це повинно відображатися і в системі захисту.

При розробці систем інформаційної безпеки необхідно враховувати передові тенденції розвитку інформаційних технологій. Характерними рисами яких є те, що вони ґрунтуються на технології клієнт/сервер, мають в своєму складі різнорідні корпоративні інформаційні системи і користуються зовнішніми сервісами, основним для яких є стек протоколів TCP/IP, а також надає аналогічні сервіси зовні.

Для захисту КС від несанкціонованого втручання в процеси їх функціонування і несанкціонованого доступу до інформації використовуються наступні методи захисту (захисні механізми):

- ідентифікація, автентифікація користувачів системи;

- розмежування доступу користувачів до ресурсів системи і авторизація користувачів;
- реєстрація і оперативне оповіщення про події, що відбуваються в системі;
- криптографічне шифрування збережених і переданих по каналах зв'язку даних;
- контроль цілісності та автентичності даних;
- резервування та створення резервних копій;
- фільтрація трафіку і трансляція адрес мережі;
- виявлення вторгнень (атак);
- виявлення і нейтралізація дій комп'ютерних вірусів;
- виявлення вразливостей системи;
- маскування і створення сурогатних об'єктів;
- страхування ризиків.

Перераховані механізми захисту можуть застосовуватися в конкретних технічних засобах та системах захисту в різних комбінаціях і варіаціях. Найбільший ефект досягається при їх системному використанні в комплексі з іншими видами заходів захисту.

Зростання загроз вторгнень викликає необхідність удосконалення підходів і методів забезпечення інформаційної безпеки в РКС, пошуку нових рішень в області створення СЗІ. Разом з традиційними засобами захисту, такими як антивіруси, міжмережні екрани і детектори вторгнень, застосовуються засоби автоматизації захисту, що включають корелятори подій, засоби автентифікації, авторизації та адміністрування, системи управління ризиками.

На сьогоднішній день широко починають використовуватися інтелектуальні СЗІ для виявлення вторгнень.

Система виявлення вторгнень отримує інформацію про комп’ютерну систему для використання діагностики стану безпеки останньої. Мета полягає в тому, щоб виявити порушення безпеки або спроби порушення, відкриті

вразливості, які можуть привести до потенціальних порушень[8].

Технології виявлення вторгнень не роблять систему абсолютною безпечною, проте практична користь СВВ незаперечна.

Головною перевагою використання СВВ є можливість аналізу неповних вхідних даних або сигналу з будь-якими перешкодами, а також проведення нелінійного аналізу подій, що відбулися у разі розподіленого зовнішнього впливу на мережу. Також такі системи здатні до навчання і можуть виявляти нові види атак.

Проведений аналіз показав що запорукою якісного виявлення аномалій є побудова ефективної системи моніторингу, тому розглянемо сучасні системи моніторингу.

2.2 Аналіз принципів побудови сучасних систем моніторингу розподілених комп’ютерних систем

Останнім часом все частіше виникають ситуації кібернетичних атак на розподілені комп’ютерних системи. Так в 2012 нафтогазова компанія Saudi Aramco, яка є власністю Саудівської Аравії, визнала, що її комп'ютерні системи піддалися кібератаці. При цьому зараженню піддалися практично всі робочі станцій її співробітників, включаючи їх суперкомп'ютер для моделювання процесів розподілу нафти при видобутку [9].

Кращим прикладом шкідливих програм, що використовуються для атак на РКС, а не на рядових користувачів є черв'як Stuxnet. Насправді, Stuxnet сам був частиною комплексу з декількох шкідливих програм, які завершують один одного з точки зору їх функціоналу і своїх цілей [10].

Як показав аналіз, хакерам вдалося отримати доступ до одних з найпотужніших суперкомп'ютерів у світі і до їхніх мереж, а цікавим фактом був не сам злом системи, а те, що злом не виявлено технічними засобами. Відповідно система моніторингу є одним з важливих компонентів системного програмного забезпечення РКС. Вона дозволяє контролювати рівень

використання ресурсів системи, а також знаходити несправності, пов'язані з роботою устаткування, що необхідно для підтримки високого ступеня надійності РКС.

Таким чином, збільшення складності структури обчислювального комплексу при його масштабуванні виникає потреба в системі, здатної самостійно оцінити стан цілісності процесу функціонування РКС.

В роботі [11] запропоновано підхід до аналізу, згідно якого були проаналізовані кілька існуючих систем моніторингу.

На сьогоднішній день існує безліч розроблених систем моніторингу. Прикладом є система Nagios - це система, розроблена для моніторингу комп'ютерних систем і мереж. Вона сканує зазначені вузли та служби, і оповіщає адміністратора в тому випадку, якщо якісь із служб припиняють (або відновлюють) свою роботу [12]. На рисунку 2.1 представлена архітектура системи.

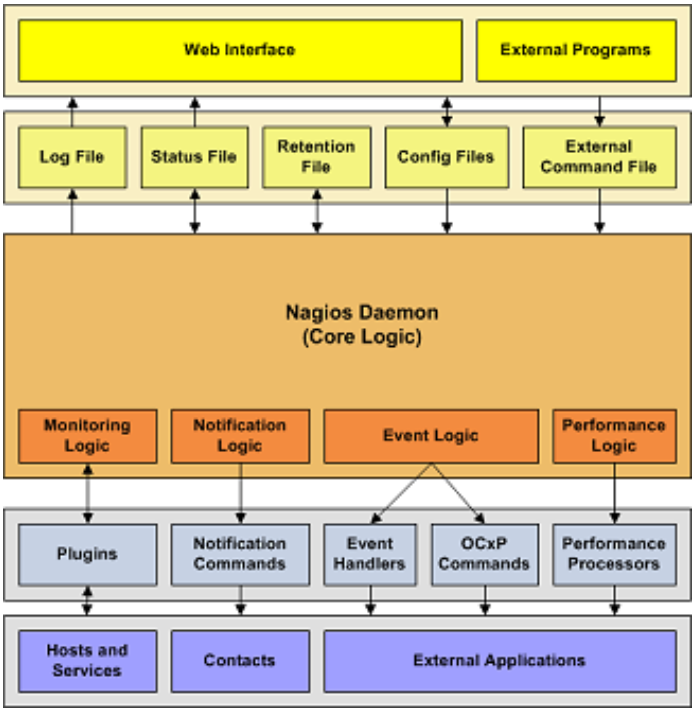

Рисунок 1.4 – Архітектура системи Nagios[27]

Недоліком такої системи є:

- погана масштабованість;
- великий інтервал між вимірами параметрів;
- усереднення даних (неможливо визначити точне значення параметрів, наприклад, місяць тому);
- відсутність засобів автоматизованого експертного аналізу даних;

Іншим прикладом подібних систем є продукт Zabbix, створений для моніторингу та відстеження статусів різноманітних сервісів комп'ютерної мережі, серверів та мережного обладнання.

Zabbix - стабільна і надійна система моніторингу зі стійкою швидкістю розвитку. Сервер працює з єдиною базою даних і незалежно від дій, з будь-якими іншими ресурсами (пам'ять, мережа, CPU), в якийсь момент часу можна зіткнутися з обмеженням I/O на диску, що використовується базою даних. Zabbix слабо підготовлений для різноманітного оточення, яке управляється системою управління конфігурацій. Він має вбудовані можливості для low-level виявлення хостів та сервісів, але вони мають свої обмеження і не мають прив'язки до системи конфігурації. Єдина можливість для подібної інтеграції - власне рішення, яке використовує API. Zabbix добре логує дії користувачів, за винятком одного сліпої плями: зміни, зроблені через API, здебільшого не логуються. UI Zabbix'a зручний і включає в себе багато можливостей. На рисунку 2.2 представлена загальна архітектура системи.

Його недоліком є погана масштабованість, низька відмовостійкість і відсутність засобів аналізу даних.

Система моніторингу, запропонована автором статті [13] за допомогою мультиагентного підходу вирішує проблеми масштабованості і відмовостійкості. Автор пропонує створити агентів, які розосереджуються по обчислювальним вузлам і збирають дані про продуктивність системи.

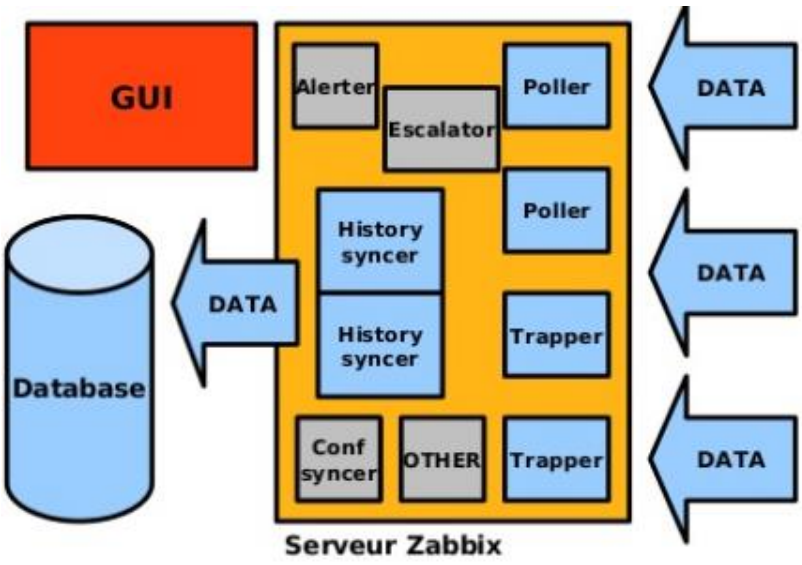

Рисунок 2.2 – Архітектура системи Zabbix[11]

На рисунку 2.3 показана пропонована архітектура системи, яка складається з 3 ключових компонентів :

1) диспетчер роботи;

2) монітор роботи;

3) журнали завдань.

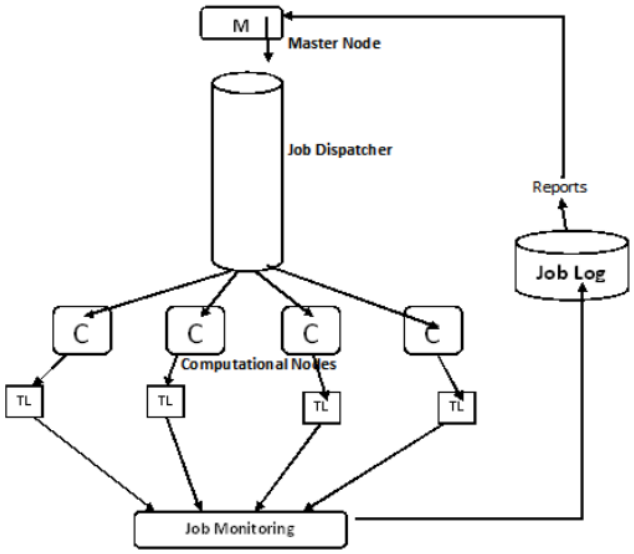

Рисунок 2.3 – Архітектура системи [12]

Недоліком такого моніторингу є те що розглядаються тільки обчислювальні вузли. Внаслідок цього аналізу даних проводиться тільки для них і це дає можливість зловмисникам провести атаку не на вузли системи, а на її мережу.

У статті [14] пропонується динамічна реконфігурована розподілена модульна система моніторингу. Тут запропоновано разом з агентним підходом, використовувати статистичний збір даних тих систем РКС, де з якихось причин немає можливості створити агента. Структура даної системи представлена на рисунку 2.3

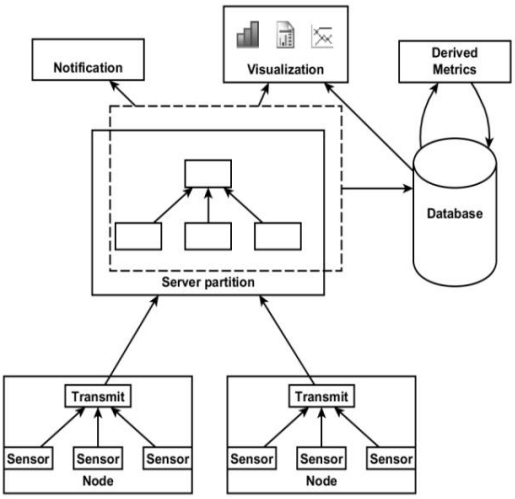

Рисунок 2.3 – Архітектура системи [13]

Недоліком такої системи є не стандартизовані дані, отримані з різних джерел, що призводить до складності їх інтелектуального аналізу.

В роботі [14] запропонований інструментальний комплекс мета моніторингу розподілених обчислювальних середовищ.

Дана система моніторингу реалізовує наступну архітектуру представлену на рисунку 2.4, що включає наступні основні компоненти:

- засоби доступу користувачів, що дозволяють взаємодіяти з системою мета моніторингу як в пакетному, так і в інтерактивному режимах;
- підсистеми рівня доступу, що здійснюють контроль прав доступу до даних, які запитує користувач і реалізують серверну частину графічного

інтерфейсу користувача;

- агент верхнього рівня, що функціонує в центральному вузлі РКС і виконує основне завдання по управлінню системою мета моніторинга;

- агенти проміжного рівня, що функціонують в проміжних вузлах і вирішальні завдання зниження навантаження на агентів верхніх рівнів;

- агенти нижнього рівня, що функціонують в вузлах РКС і здійснюють збір і первинну обробку даних про стан вузлів;

- підсистема децентралізованого зберігання даних, що надає функції для роботи з даними для агентів різних рівнів.

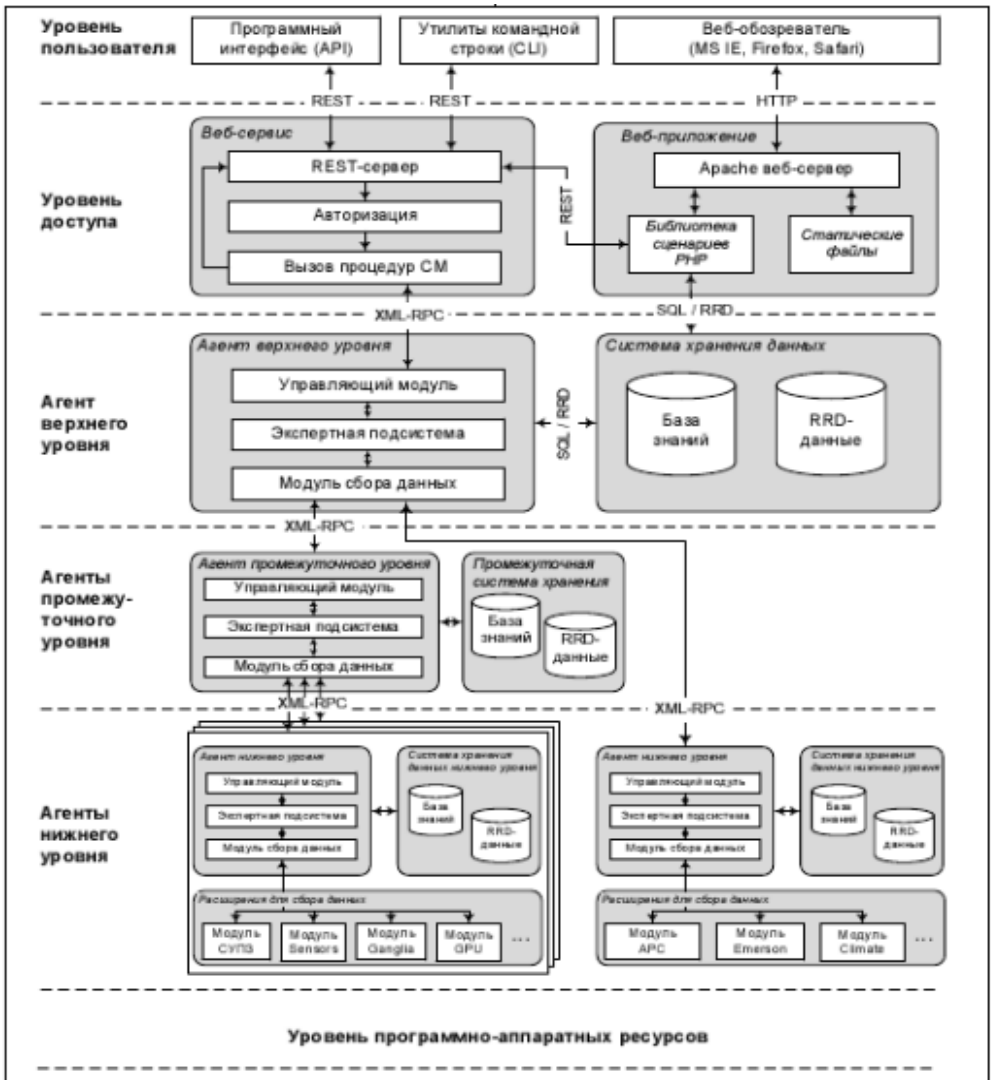

Рисунок 2.4 – Архітектура системи метамоніторингу [14]

Її недолік полягає в тому, що для інтелектуального аналізу даних

використовується експертна система, розроблена з використанням оболонки CLIPS. У даній експертній системі відсутній механізм самонавчання. Це призводить до того, що система не може виявити нові аномалії.

Автор роботи [15] спільно з Nagios використовує нейронні мережі для визначення аномалій, що вирішує проблему навчання, але не вирішує інших недоліків даної системи.

Кожен з цих недоліків може привести до порушень безпеки функціонування. Основною метою роботи є створення такої підсистеми моніторингу, яка на основі даних про роботу РКС могла б оцінити цілісність функціонування обчислювального процесу РКС. Саме цим зумовлена необхідність досліджень в даному напрямку.

Виходячи з того що після проведення моніторингу базовим процесом встановлення кібернетичного впливу є процес виявлення аномалій.

3 ФОРМУЛЮВАННЯ ПІДХОДІВ ЩОДО ВИРІЩЕННЯ ЗАВДАНЬ КЛАСИФІКАЦІЇ АНОМАЛІЙ

3.1 Підходи щодо вирішення завдань класифікації аномалій

При вирішенні складних завдань класифікації, регресії, прогнозування часто виявляється, що жоден з алгоритмів не забезпечує бажаної якості відновлення залежності. У таких випадках має сенс будувати композиції алгоритмів, в яких помилки окремих алгоритмів взаємно компенсуються. Найбільш загальне визначення алгоритмічної композиції дається в алгебраїчному підході Ю. І. Журавльова [16]. Поряд з множиною об'єктів і множиною відповідних їм значень цільової функції вводиться допоміжна множина, яка називається простором оцінок. Розглядаються алгоритми, в яких функція, що називається алгоритмічним оператором, встановлює відповідність між множиною об'єктів і простором оцінок, а функція, що називається вирішальним правилом, встановлює відповідність між простором оцінок і множиною значень цільової функції. Таким чином, розглянуті алгоритми мають вигляд суперпозиції алгоритмічного оператора і вирішального правила. Багато алгоритми класифікації мають саме таку структуру: спочатку обчислюються оцінки приналежності об'єкта класам, а потім вирішальне правило переводить ці оцінки в номер класу. На рисунку 3.1 представлена загальна схема алгоритмічної композиції. Значенням оцінки може бути ймовірність приналежності об'єкта класу, відстань від об'єкта до розділяємої поверхні, ступінь впевненості класифікації та інше.

Існує кілька найбільш відомих методів об'єднання базових алгоритмів в композиції: голосування, зважене голосування, суміш експертів. Ці методи часто застосовуються, коли базові алгоритми істотно відрізняються один від одного. У випадках, коли необхідно побудувати композицію використовуючи один базовий алгоритм, широко застосовується bagging або boostrap

aggregation [17, 18].

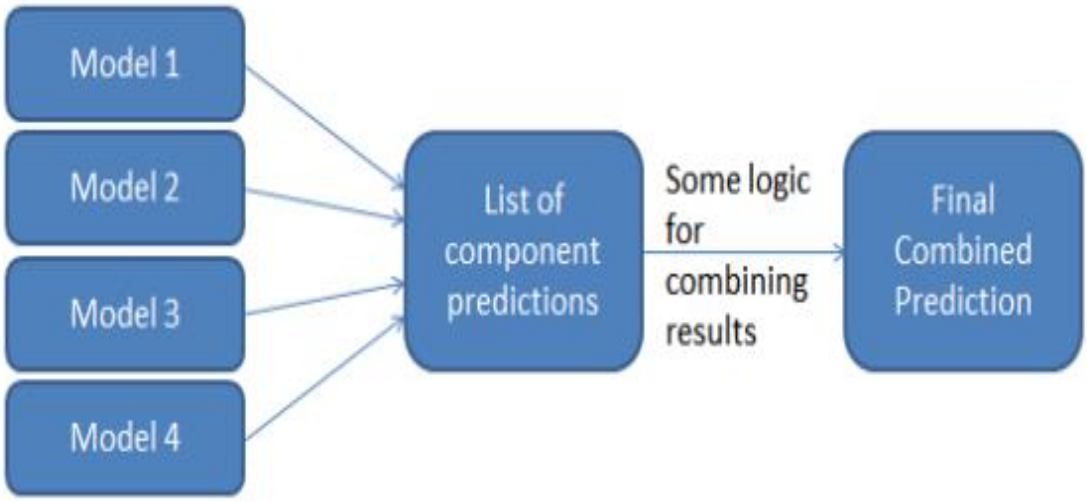

Рисунок 3.1 – Загальна схема алгоритмічної композиції

Ідея bagging полягає в тому, що базовий алгоритм багаторазово навчається на випадкових підвибірках з повтореннями з навчальної вибірки. Такий метод генерації підвибірок прийнято називати bootsrap. Схожим на беггінг методом є метод випадкових підпросторів random subspace method, RSM [19]. Його ідея полягає в створенні варіативності при навчанні за допомогою вибору випадкових підмножин ознак. Широко відомим прикладом використання беггінга і RSM є RandomForest [20].

Іншим відомим способом об’єднання базових алгоритмів в ансамбль є бустінг. Ідея бустінга складається в жадібному виборі чергового алгоритму для додавання в композицію так, щоб він найкращим чином компенсував наявні на цьому кроці помилки. Широко відомі приклади бустінга - AdaBoost [21] і Gradient boosting [22].

Стекінг (Stacked generalization) був вперше запропонований Д. Волпертом в 1992 році в роботі [23] в досить загальному вигляді. Основна ідея стекінг полягає в використанні базових класифікаторів для отримання прогнозів і використанні їх як ознак для деякого "узагальнюючого" алгоритму.

Іншими словами, основною ідеєю стекінгу є перетворення початкового простору ознак в новий простір, точками якого є передбачення базових алгоритмів. Пропонується спочатку вибрати набір пар довільних підмножин з навчальної вибірки, потім для кожної пари навчити базові алгоритми на першому підмножині і передбачити ними цільову змінну для другого підмножини. Передбачені значення і стають об'єктами нового простору. Зокрема, автором розглядається випадок вибору всіляких пар підмножин, в якому друга підмножина складається з єдиного об'єкта, а перша підмножина з усієї навчальної вибірки крім цього об'єкта (leave-one-out). Очевидно, що такий спосіб дозволяє перевести кожну точку початкового простору ознак в точку нового простору. Автор узагальнює ідею стекінг тим, що пропонує, навчаючи базові класифікатори (першого рівня) над мета ознаками (першого рівня), отримувати мета ознаками другого рівня і так далі.

3.2 Дослідження принципів послідовно навчання базових алгоритмів

Процедури голосування мають безліч форм, таких як голосування абсолютної більшості, мажоритарні виборчі системи, що схвалють голосування і багато інших.

Процес послідовного навчання базових алгоритмів, використовується найчастіше при побудові композицій. Розглянемо спочатку цей процес в найбільш загальному вигляді, який показано в алгоритмі представленому на рисунку 3.2.

Параметри X^l, Y^l, – навчальна вибірка, μ – метод навчання базових класифікаторів, T – максимальна кількість алгоритмів в композиції, $F(b_1, \ldots, b_t)$ – алгоритмічна композиція, $M(W^l)$ – функція модифікація ваг класифікатора.

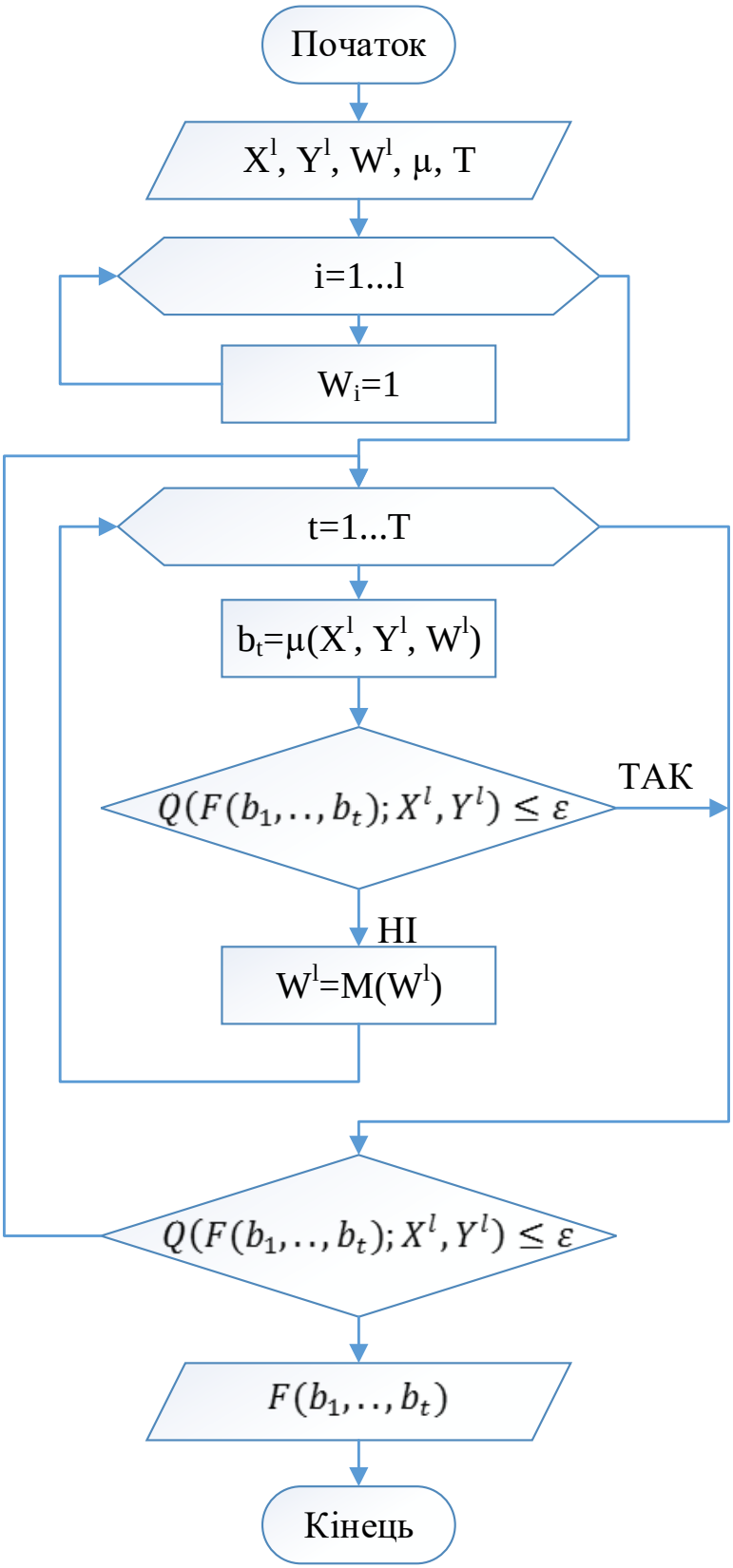

Рисунок 3.2 – Побудова алгоритмічної композиції шляхом послідовного навчання базових алгоритмів

На першому кроці за допомогою стандартного методу навчання μ

будується перший базовий алгоритм b_1. Якщо його якість задовольняє, то подальша побудова композиції непотрібна, і процес на цьому завершується. В іншому випадку алгоритм b_1 фіксується і будується другий алгоритм b_2, при одночасній оптимізації коригуючої операції F. На t-му кроці базовий алгоритм b_t і коригуючи операція F оптимізуються при фіксованих b_1, ..., b_{t-1}:

$$b_1 = arg \max_b Q(b; X^l, Y^l); \quad (3.1)$$

$$b_2 = arg \max_{b,F} Q(F(b_1, b); X^l, Y^l); \quad (3.2)$$

$$b_t = arg \max_{b,F} Q(F(b_1, .., b_t); X^l, Y^l); \quad (3.3)$$

У багатьох випадках для вирішення задачі (3.3) вдається пристосувати стандартні методи навчання, які вирішують задачу більш простого виду (3.1). Для цього на вхід стандартного методу навчання $\mu(X^l, Y^l, W^l)$ подаються модифіковані вектори ваг W^l і відповідей Y^l. Конкретний спосіб модифікації залежить від типу задачі класифікація або регресія і виду коригуючої операції F. Для кожного окремого випадку формули перерахунку ваг і відповідей виводяться окремо. Модифікація ваг, як правило, зводиться до збільшення ваги у найбільш «важких» об'єктів, на яких частіше помилялися попередні базові алгоритми, а модифікація відповідей - до апроксимації похибка обчислень виду y_i-$a(x_i)$ замість апроксимації вихідних відповідей y_i.

Базовий алгоритм b_t, оптимальний на t-му кроці, перестає бути оптимальним після додавання наступних алгоритмів. Процес (3.1)-(3.3) можна узагальнити, чергуючи додавання нових алгоритмів з пере налаштуванням попередніх алгоритмів:

$$b_k = arg \max_{b,F} Q(F(b_1, .., b_{k-1}, b, b_{k+1} b_t); X^l, Y^l), 1 \le k < t \quad (3.4)$$

За способами вирішення ця задача мало чим відрізняється від задачі побудови останнього базового алгоритму (3.3).

Критерії зупинки можуть використовуватися різні, в залежності від специфіки задачі, можливо також спільне застосування декількох критеріїв:

- побудовано задану кількість базових алгоритмів *T*;
- досягнуто задана точність на навчальній вибірці:

$$Q(F(b_1,..,b_t);X^l,Y^l) \leq \varepsilon \tag{3.5}$$

- досягнуту точність на контрольній вибірці X^k не вдається поліпшити протягом останніх d кроків: $t - t^* > d$, де d - параметр алгоритму;

$$t^* = arg \max_{s=1,...,t} Q\,(F(b_1,..,b_s);X^k,Y^k); \tag{3.6}$$

Виконання цього критерію вважається ознакою перенавчання. В якості остаточного рішення береться композиція, побудована на t* -му кроці.

Послідовне побудову базових алгоритмів найпростіше реалізується, коли коригувальна операція не має власних параметрів, що настроюються. До таких методів відноситься голосування за більшістю і по старшинству.

Послідовна побудову базових алгоритмів найпростіше реалізується, коли коригуюча операція не має власних параметрів, які можна налаштовувати. До таких методів відноситься голосування по більшості і по старшинству.

3.3 Дослідження принципів комбінації алгоритмів методом голосування по більшості

Розглянемо задачу класифікації з двома класами $Y = \{-1, +1\}$, простором оцінок $R = \mathbb{R}$ і вирішальним правилом $C(b)=sign(b)$. Як коригувальної операції візьмемо просте голосування. Розглянемо функціонал якості композиції a,

який дорівнює числу помилок при навчанні:

$$t^* = arg \max_{s=1,\dots,t} Q\left(F(b_1,..,b_s); X^k, Y^k\right); \quad (3.6)$$

На рисунку 3.3 представлено алгоритм побудови композиції для голосування по більшості.

Параметри X^l, Y^l – навчальна вибірка, μ – метод навчання базових класифікаторів, T – максимальна кількість алгоритмів в композиції, $F(b_1,\dots, b_t)$ – алгоритмічна композиція, l_l – довжина навчальної вибірки, $Sort(X^l,M_i)$ функція впорядкування X^l за значеннями відступів M_i.

Якщо відступ негативний, $M_{it} < 0$, то композиція перших t базових алгоритмів допускає помилки на об'єкті x_i. Щоб компенсувати помилки композиції, будемо навчати базовий алгоритм b_t+1 не на всій вибірці X^l, а тільки на об'єктах з найменшими значеннями M_{it}. Або, будемо мінімізувати функціонал $Q(b_{t+1}; W^l)$ з вагами об'єктів $w_i = [M_{it} \leq M_0]$.

Обирати параметр M_0 слід так, щоб в навчальну вибірку потрапило не надто мало об'єктів (інакше будуть будуватися базові алгоритми занадто низької якості), але і не дуже великі (інакше будуть будуватися майже однакові алгоритми). Тому замість M_0 зручніше ввести параметр довжини навчальних підвибірок l_1 і підбирати його оптимальне значення.

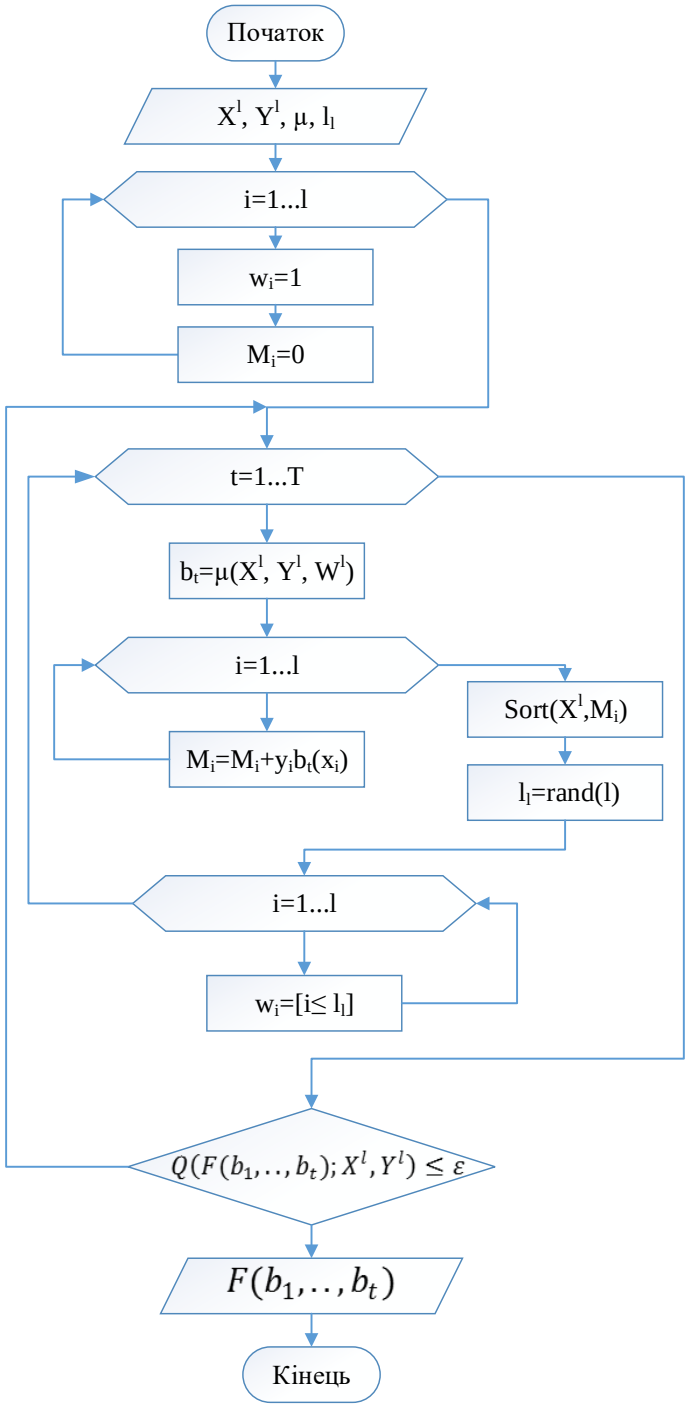

Рисунок 3.3 – Алгоритм побудови композиції для голосування по більшості

Проведений аналіз показав, що спектр методів досить широкий, від класичного статистичного аналізу до апарату штучних нейронних мереж. Але жоден метод не розглядає роботу з великим числом категоріальних ознак. А також в них не розглядається визначення різних типів кібернетичних атак.

4 РОЗРОБКА МЕТОДУ КЛАСИФІКАЦІЇ АНОМАЛІЙ

4.1 Постановка задачі класифікації стану мережі

Класична задача класифікації об'єктів представлена в статті [24].

Потрібно розробити алгоритм, який зможе по ознакам опису нового об'єкта видати значення його цільової зміної.

Набір даних, який використовується в роботі для побудови моделі, був змодельований і отриманий в мережі обміну даними кластерного суперкомп'ютера. За основу брався набір параметрів, представлений на змаганні по машинному навчання KDD cup 2009 з додаванням параметрів для моніторингу сховища даних.

У таблицях 4.1-4.4 наведено приклад вихідних даних для задачі класифікації.

Таблиця 4.1 – TCP параметри

Параметр	Опис
Duration	Тривалість з'єднання(секунди)
protocol _ type	Протокол транспортного рівня
Service	Сервіс прикладного рівня
number of data bytes source - destination	Потік, що входить, байт
number of data bytes destination - source	Вихідний потік, байт
Flagstatus	Прапори, встановлені в заголовку TCP- пакету.
Land	Адреси співпадають, 0 інакше
wrong _ fragment	Число неправильних фрагментів
urgentpackets	Наявність термінових даних в пакеті(прапор URG).

Таблиця 4.2 – Характеристики сеансу

Параметр	Опис
Hot	Число «hot» індикаторів
num_failed_logins	Число невдалих спроб входу
logged_ n	Успішний вхід
num_compromised	Доступ з адміністративними повноваженнями
root_shell	Число спроб доступу з правами адміністратора
num_root	Число операцій з файлами контролю доступу
num_file_creations	Кількість операцій створення файлу
num_access_files	Кількість операцій на файлах управління доступом
num_compromised	Кількість скомпрометованих статусів
is_hot_login	Приналежність користувача до «hot» списку
is_guest_login	Чи є користувач гостем

Таблиця 4.3 – Статистика з'єднання за 2 секунди

Параметр	Опис
Count	Число з'єднань із співпадаючим хостом
serror_rate	% з'єднання з помилкою "SYN"
Service	% з'єднань з помилкою "REJ" / % з'єднань з однаковим початковим портом
rerror_rate	% з'єднань з однаковим сервісом
same_srv_rate	% з'єднань з різним сервісом
diff_srv_rate	Число з'єднань із співпадаючим сервісом
srv_count	Число з'єднань із співпадаючим сервісом
srv_serror_rate	% з'єднань з помилкою "REJ"
srv_rerror_rate	% з'єднань з помилкою
srv_diff_host_rate	% з'єднань з хостами, що розрізняються

Таблиця 4.4 – Характеристики файлової системи кластера (Lustre)

Параметр	Опис
num _ exports	Кількість експорту на MDT - це клієнти, у тому числі інші сервери Lustre
Stats	Перераховані клієнтські з'єднання по NID.
lock _ count	Кількість блокувань
pool.granted	Luster розподілений менеджер блокування(ldlm) надав блокування
grant _ rate	ldlm заблокований рівень відміни, що має назву 'GR'
cancel _ rate	Ldlm заблокований рівень відміни, що має назву 'CR'

4.2 Метод класифікації стану мережі на основі модифікованого алгоритму стекинга

Стекинг використовує концепцію мета-навчання, тобто намагається навчити кожен класифікатор, використовуючи алгоритм, який дозволяє виявити кращу комбінацію виходів базових моделей .

Послідовність роботи цього алгоритму в спрощеному виді складається з наступних етапів:

1) на вхід алгоритму подати повчальну вибірку $X=\{x_1.,x_n\}$ і множина базових алгоритмів класифікації $A=\{a_1,...,a_m\}$;

2) множину X розбити на дві підмножини Xa і Xb, що не перетинаються;

3) навчити множину базових класифікаторів A на підмножині Xa;

4) тестувати базові класифікатори A на підмножині Xb, as: Xb → Yb;

5) використовуючи множина Yb як вхідні дані для мета-алгоритма, а істинні значення цільової змінної як вихідні значення навчити мета-алгоритм.

Алгоритм роботи класичного стекинга представлений на рисунку 4.1.

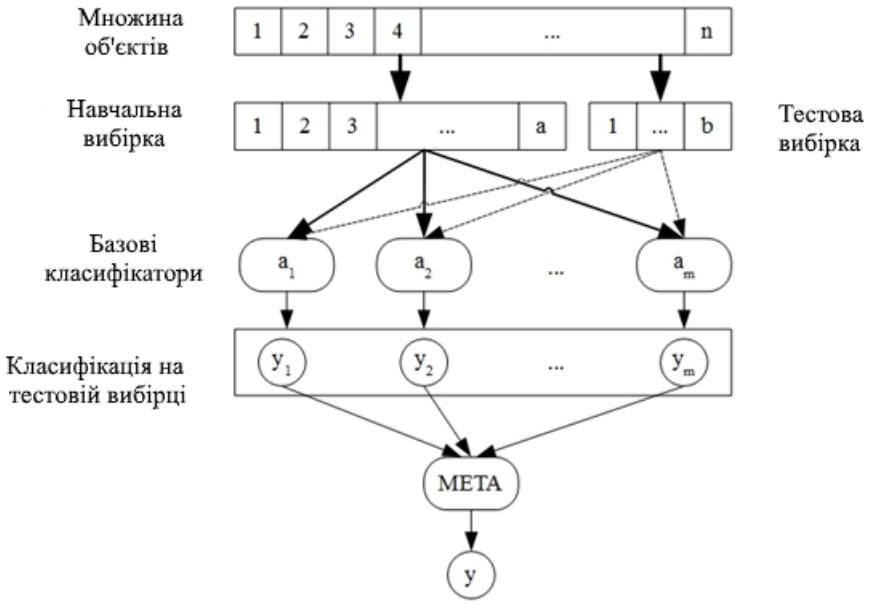

Рисунок 4.1 – Схема класичного стекинга

Недоліком цього алгоритму є те, що базові алгоритми навчаються не на усій множині об'єктів. Це у свою чергу призводить до того, що в навчальу підмножини X^a може не потрапити жодного об'єкту класу k і виникне недообучання множини базових алгоритмів класифікаторів A, а як слідство недообучання алгоритму стекинга в цілому.

У роботі пропонується модифікований алгоритм стекинга. Функціонування цього алгоритму описується наступним кортежем:

$$(A_{ij}, Out_{ij}, In_{ij}, x_n, A_{meta}), \tag{4.1}$$

де $A_{ij-1} : In_{ik} \to Out_{ij}$ здатний класифікувати довільний об'єкт множини $\mathrm{In_j}$.

$$A_{meta} : In_{ik} \to Y, \tag{4.2}$$

здатний класифікувати довільний об'єкт множени In_k.

x - множина вхідних значень;

k - кількість рівнів стекинга;

In_{ik} - множина вхідних об'єктів і- го алгоритму на j- ом рівні;

Out_{ij} - множина вихідних об'єктів і- го алгоритму на j- ом рівні;

i - номер алгоритму на рівні;

j - номер рівня стекинга;

Y - множина цільових змінних;

Алгоритму роботи якого складається з наступних етапів :

1) на вхід алгоритму подати повчальну вибірку $X=\{x_1, ...,x_n\}$ і множина базових алгоритмів класифікації $A=\{a_1, ...,a_m\}$;

2) множину X розбити на K підмножин, що пересікаються. Шляхом рівномірної вибірки L об'єктів з поверненням. Кожна підмножина будується з використанням різних об’єктів вихідної вибірки X. Приблизно 37% об'єктів залишаються поза підмножиною і не використовуються при побудові K - ї підмножини;

3) навчити множину базових класифікаторів A K- го рівня стекинга на підмножині K;

4) тестувати базові класифікатори K- го рівня на множині об'єктів які не увійшли в K - у підмножину;

5) використовуючи множину об'єктів які не увійшли в K-у підмножину, як вхідні дані для мета-алгоритма, а істинні значення цільової змінної як вихідні значення навчити мета-алгоритм.

Робота модифікованого алгоритму стекинга представлена на рисунку 4.2.

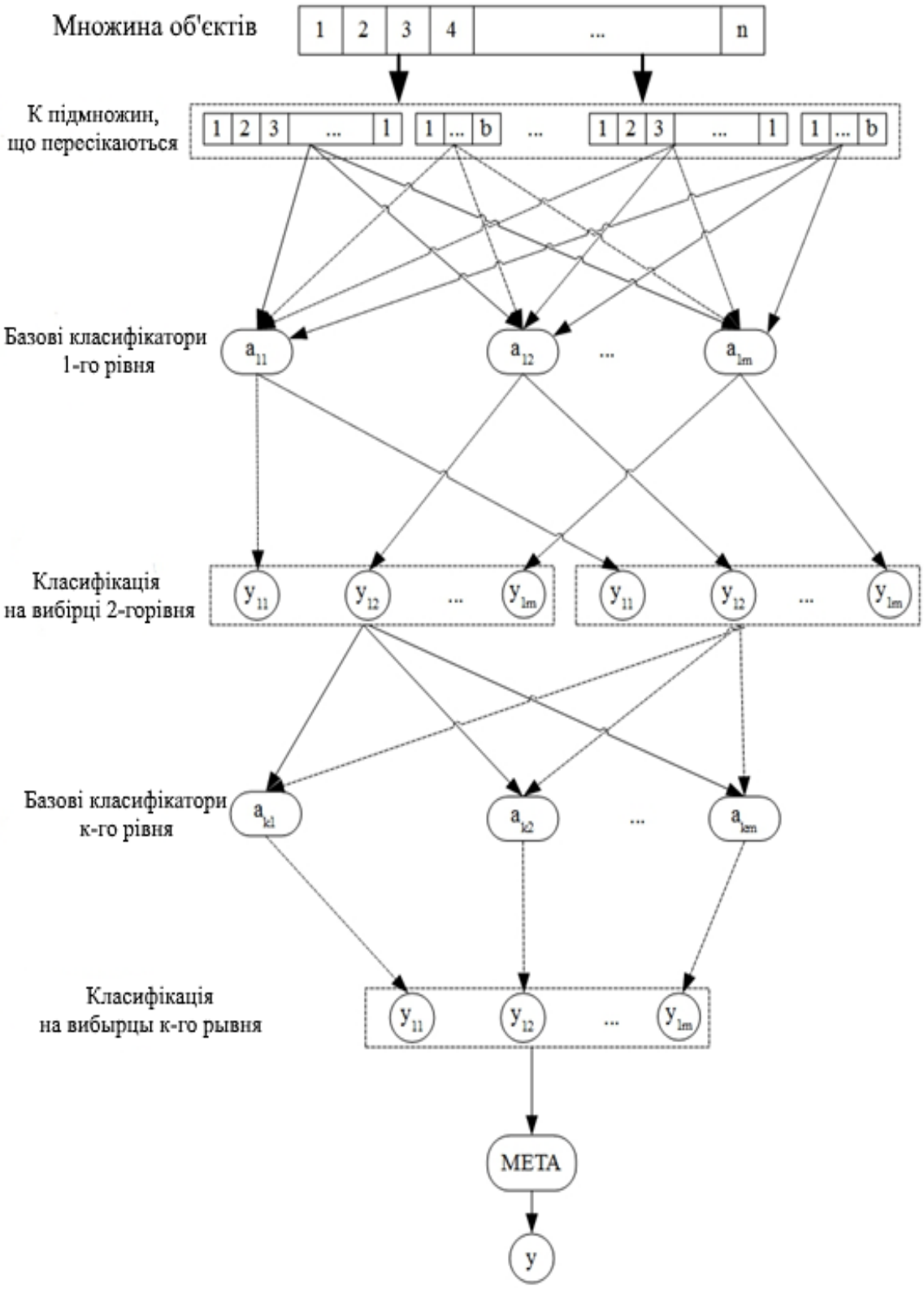

Рисунок 4.2 – Схема модифікованого стекінга

Цей алгоритм дозволяє позбавиться від не до навчання і дозволить застосовувати його на навчальних вибірках невеликих розмірів.

4.3 Результати дослідження методів класифікації стану мережі

Модифікований алгоритм стекинга, дозволяє використати менше число об'єктів навчальної вибірки. Також проводить до поетапного зменшення простору ознак для мета-классификатора з меншою мірою кореляції.

Проведена оцінка ефективності запропонованого алгоритму з роботою базових класифікаторів і класичного стекинга. Для проведення порівняльного аналізу використовувалися дані з чемпіонату по машинному навчанню KDD 2009 і дані отримані при моніторингу мережевої інфраструктури учбового дата-центра розгорнутого на основі мережевої файлової системи Lustre.

У таблиці 4.5 представлені класи атак на мережі що знаходяться в навчальній і тестовій вибірці KDD 2009 .

Таблиця 4.5 – Класи атак в навчальній і тестовій вибірці

Навчальна вибірка	Тестова вибірка
back, buffer_overflow, ftp_write, guess_passwd, imap, ipsweep, land, loadmodule, multihop, neptune, nmap, phf, pod, portsweep, rootkit, satan, smurf, spy, teardrop, warezclient, warezmaster, normal	guess_passwd, imap, ipsweep, land, loadmodule, multihop, neptune, nmap, phf, pod, portsweep, rootkit, satan, smurf, spy, teardrop, warezclient, warezmaster apache2, httptunnel, mailbomb, mscan, named, perl, processtable, ps, saint, sendmail, snmpgetattack, snmpguess, sqlattack, udpstorm, worm, xlock, xsnoop, xtermbuffer_overflow, Neptune, warezmaster, smurf, normal

Таблиця 4.6 – Класи атак в навчальній і тестовій вибірці

Навчальна вибірка	Тестова вибірка
back, neptune, pod, spy, normal	back, neptune, pod, smurf, spy, teardrop, normal

Для проведення експерименту використовувалися такі базові класифікатори:

- класифікатор kNN;
- наївний класифікатор Байеса;
- дерева класифікації;
- SVM;

Як мета-классификатора використовувався багатошаровий персептрон.

Перший етап експерименту включає підготовку даних для навчання базових класифікаторів. Оскільки в експерименті використовуються такі алгоритми машинного навчання, як kNN і SVM, які чутливі до масштабування даних, то для чисельних ознак використовуватимемо нормалізація по методу мінімакса.

Для категоріальних ознак було використано кодування, розглянуте в статті [22], суть якої полягає в наступному.

Нехай F - деяка множена дійсних функцій, в якій для довільного натурального числа k рівна одна функція від k змінних. При цьому усі функції симетричні, тобто для будь-якої функції (від k змінних з F для будь-якої перестановки (. Прикладом таких великих кількостей може бути множена сум середніх арифметичних, максимумів і так далі

Для кодування значення f_{jj} -ої категоріальної ознаки вибираємо множену об'єктів з навчання з таким значенням:

$$I = \{t \in \{1,2,...,l\} \mid f_{tj} - f_j\}, \tag{4.3}$$

вибираємо дійсну ознаку, відносно якої кодуватимемо, наприклад s - й, і кодуємо значення f_j значенням відповідної функції з F, тобто функції від $|I|$ змінних від значень f_{is}, Наприклад, кодування протоколу здійснювалося його заміною на арифметичне середнє значення тривалості сеансу для цієї категорії.

Другий етап — це навчання базових алгоритмів, стандартного і

модифікованого стекинга.

Після чого було проведено експериментальне дослідження ефективності роботи базових класифікаторів, класичного і модифікованого стекинга, результати якого приведені в таблицях 4.7-4.8.

Таблиця 4.7 – Результати експерименту на датасете KDD 2009

Кількість випробувань	Кількість вірних рішень		Кількість невірних рішень			
			Помилки I роду		Помилки II роду	
	Навч-на вибірка	Тестова вибірка	Навч-на вибірка	Тестова вибірка	Навч-на вибірка	Тестова вибірка
Наївний класифікатор Байеса	80,59	64,90	15,68	23,40	3,73	11,70
Класифікатор kNN	85,40	70,40	10,60	19,97	4,00	9,63
SVM	84,30	66,47	10,60	19,93	5,10	13,60
Дерева класифікації	85,70	72,80	8,10	15,60	6,20	11,60
Стандартний стекинг	86,64	71,38	3,09	15,82	10,27	12,8
Модифікований стекинг	92,01	84,19	4,7	10,49	3,29	5,32

Таблиця 4.8 – Результати експерименту на датасете отримані при моніторингу обчислювальної мережі кластера

Кількість випробувань	Кількість вірних рішень		Кількість невірних рішень			
	Навч-на вибірка	Тестова вибірка	Помилки I роду		Помилки II роду	
			Навч-на вибірка	Тестова вибірка	Навч-на вибірка	Тестова вибірка
Наївний класифікатор Байеса	79,45	62,45	16,8	25,6	3,75	11,95
Класифікатор kNN	82,8	67,74	12,8	20,2	4,4	12,06
SVM	80,13	66,47	12,02	19,93	7,85	13,6
Дерева класифікації	81,23	69,3	11,07	17,23	7,7	13,47
Стандартний стекинг	82,44	68,8	5,16	17,82	12,4	13,38
Модифікований стекинг	92,30	85,46	4,7	10,49	3,00	4,05

ВИСНОВКИ

У кваліфікаційній роботі вирішена актуальна наукова задача покращення показників виявлення аномалій функціонування розподілених інформаційних систем в умовах кібернетичних впливів зовнішнього та внутрішнього середовища шляхом побудови моделей і методів на основі технологій інтелектуального аналізу даних.

Основні результати виконаної роботи полягають в наступному:

1) проведено аналіз існуючих методів і засобів моніторингу розподілених інформаційних систем, який виявив відсутність надання цілісної оцінки стану функціонування РІС;

2) розглянуто особливості архітектури розподілених комп'ютерних систем, що дозволило виявити базові компоненти розподілених комп'ютерних систем, які потребують постійного моніторингу їх стану;

3) визначено множину параметрів для оцінки стану кожного елемента системи, що дає можливість для подальшої оцінки функціонування системи в цілому та скоротити час навчання ансамблю класифікаторів на 30,79%;

4) розроблено метод виявлення аномалій в мережі на основі стекінгу методів інтелектуального аналізу даних, який на відміну від існуючих алгоритмів виявлення аномалій дозволяють підвищити точність виявлення до 92,7% і знизити кількість помилкових другого роду до 4,05% за рахунок використання різнорідних методів інтелектуального аналізу даних і до навчання системи на даних підчас роботи системи моніторингу.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Мартовицький В. О. Класифікація методів виявлення аномалій в інформаційних системах / В. О. Мартовицький, І. В. Рубан, С. О. Партика. // Системи озброєння і військова техніка. – 2016. – №3. – С. 100–105.

2. Martovytskyi V. Designing a monitoring model for cluster super-computer / V. Martovytskyi, I. Ruban, N. Lukova-Chuiko. // Eastern-European Journal of Enterprise Technologies. - 2016. - №84. - Pp. 32-37.

3. Martovytskyi V. Approach to Classifying the State of a Network Based on Statistical Parameters for Detecting Anomalies in the Information Structure of a Computing System / V. Martovytskyi, I. Ruban, N. Lukova-Chuiko. // Cybernetics and Systems Analysis. - 2018. - №54. - Pp. 302-309.

4. Мартовицкий В. Модель мультиагентной системы сбора и хранения информации / В. Мартовицкий, И. Рубан. // Системы управления, навигации и связи. - 2017. - №6. - С. 150-153.

5. Відбір параметрів моніторингу мережної інфраструктури для класифікації стану мережі / В. О.Мартовицький, І. В. Рубан, О. В. Сєвєрінов, О. В. Бологова. // Сучасні інформаційні системи. – 2018. – №4. – С. 5–10.

6. Денисенко М.П. Інформаційне забезпечення ефективного управління підприємством / М. П. Денисенко, І. В. Колос // Економіка та держава. – 2006. – № 7. – С. 19 – 24.

7. Кількість кіберзлочинів в Україні збільшується на 2,5 тисячі на рік [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: https://dt.ua/UKRAINE/kilkist-kiberzlochiniv-v-ukrayini-zbilshuyetsya-na-2-5-tisyachi-na-rik-266179_.html.

8. Лєнков С. В. Шляхи підвищення захисту авторського права за допомогою використання цифрових водяних знаків / С. В. Лєнков, П. А. Шкуліпа, В. І. Прухніцький. // Збірник наукових праць Військового інституту Київського національного університету імені Тараса Шевченка. – 2017. – №56.

– С. 33–40.

9. Bronk, C. The cyber attack on Saudi Aramco / Bronk C., Tikk–Ringas, E. // Survival. – 2013. – Т. 55, №. 2. – С. 81–96.

10. Knopová, M. The Third World War? In The Cyberspace. Cyber Warfare in the Middle East / Knopová, M. // Acta Informatica Pragensia. – 2014. – Т. 3, № 1. – С. 23–32.

11. Kora, A. D. Nagios based enhanced IT management system / Kora, A. D., Soidridine, M. M. // International Journal of Engineering Science and Technology (IJEST) – 2012. – Т.4, №. 4. – P. 1199–1207.

12. Cigala, V. Job–Oriented Monitoring of Clusters / Cigala, V. //International Journal on Computer Science and Engineering. – 2011. – Т. 3. – №. 3.

13. Stefanov, K. Dynamically Reconfigurable Distributed Modular Monitoring System for Supercomputers (DiMMon) / Stefanov, K. //Procedia Computer Science. – 2015. – Т. 66. – P. 625–634.

14. Сидоров, И. А. Инструментальный комплекс метамониторинга распределенных вычислительных сред / Сидоров, И. А., Опарин, Г. А., Скоров, В. В //Параллельные вычислительные технологии. – 2014. – С. 159–167.

15. Tarasov, A. G. Integration of computing cluster monitoring system / Tarasov, A. G. //Proc. of the First Russia and Pacific Conference on Computer Technology and Applications (RPC 2010). – 2010. – С. 221–224.

16. Журавлёв Ю. И. Об алгебраическом подходе к решению задач распознавания или классификации // Проблемы кибернетики. 1978. – С. 25-45.:

17. SPMoE: a novel subspace-projected mixture of experts model for multi-target regression problems / [E. Hadavandi, S. Jamal, H. Yoichi та ін.]. // Soft Computing. – 2016. – №5. – Pp. 2047–2065.

18. Onan A. assifier and feature set ensembles for web page classificatio / Onan. // Journal of Information Science. – 2016. – №42. – Pp. 150–165.

19. Baskin I. I. Bagging and boosting of classification models / I. I. Baskin. // Tutorials in Chemoinformatics. – 2017. – Pp. 241–247.

20. Belgiu M. Random forest in remote sensing: A review of applications and future directions / M. Belgiu, L. Drăguţ. // ISPRS Journal of Photogrammetry and Remote Sensing. – 2016. – №114. – Pp. 67–81.

21. Liu H. Comparison of four Adaboost algorithm based artificial neural networks in wind speed predictions / H. Liu. // Energy Conversion and Management. – 2015. – №92. – Pp. 67–81.

22. Xgboost: extreme gradient boosting / [T. Chen, T. He, M. Benesty та ін.]. // R package version 0.4-2.. – 2015. – Pp. 1–4.

23. Wolpert D. H. Stacked generalization/ D. H. Wolpert //Neural networks. – 1992. – Т. 5. – №. 2. – Pp. 241-259.

24. Вапник В. Н. Теория распознавания образов. / В. Н. Вапник, А. Я. Червоненкис //Статистические проблемы обучения. – 1974. – С. 23-25