УДК 004.056
КП
№ держреєстрації 0115U000686
Інв. №

Міністерство освіти та науки України
Сумський державний університет
(СумДУ)
40007, м. Суми, вул. Римського-Корсакова, 2
тел. (0542) 33-54-79 факс (0542) 33-54-79

ЗАТВЕРДЖУЮ
Проректор з наукової роботи
д.ф-м.н., професор

_____А.М. Чорноус

ЗВІТ
ПРО НАУКОВО-ДОСЛІДНУ РОБОТУ
«ІНТЕЛЕКТУАЛЬНА СИСТЕМА КЕРУВАННЯ НАВАНТАЖЕННЯМ І РЕСУРСАМИ РОЗПОДІЛЕНОГО ОБЧИСЛЮВАЛЬНОГО СЕРЕДОВИЩА З ПІДВИЩЕНОЮ ІНФОРМАЦІЙНОЮ БЕЗПЕКОЮ»

(заключний)

Начальник НДЧ
к.ф.-м.н., с.н.с. Д.І. Курбатов

Керівник НДР
д.т.н., професор А.С. Довбиш

2016

Рукопис завершено 15 грудня 2016 р.
Результати цієї роботи розглянуто науковою радою СумДУ,
протокол від 2016.12.23. р. №4

СПИСОК АВТОРІВ

Керівник НДР, д.т.н, професор	(15.12.2016)	А.С. Довбиш (вступ, висновки)
к.т.н., н.с.	(15.12.2016)	Шелехов І.В.(підрозділ 3.3)
к.т.н., н.с.	(15.12.2016)	Москаленко В.В.(підрозділ 1.3, розділ 2, підрозділи 3.1, 3.5)
м.н.с., аспірант	(15.12.2016)	Москаленко А.С.(підрозділи 2.2, 3.4)
пров.інж.	(15.12.2016)	Токаренко Г.В.(підрозділ 1.1)
к.т.н., с.н.с.	(15.12.2016)	Тиркусова Н.В.(підрозділ 1.2)
к.т.н., с.н.с.	(15.12.2016)	Бабій М.С.(підрозділ 2.1)
студент	(15.12.2016)	Болгов С. О.(підрозділ 3.4)
студент	(15.12.2016)	Литюга М.А.(підрозділ 2.3)
студент	(15.12.2016)	П`ятаченко В.Ю.(підрозділ 3.2)

РЕФЕРАТ

Звіт про НДР: 166 с., 23 рис., 3 табл., 81 джерело.

Об’єкт дослідження – слабо формалізований процес прийняття рішень при керуванні ресурсами розподіленого обчислювального середовища.

Предмет дослідження – моделювання, оцінка функціональної ефективності, оптимізація та класифікаційне прогнозування функціональної ефективності системи керування ресурсами розподіленого обчислювального середовища.

Мета роботи – підвищення функціональної ефективності інтелектуальної системи керування розподілом ресурсів інформаційно-телекомунікаційного середовища за умов апріорної невизначеності.

В заключному звіті виконано аналіз сучасного стану і тенденцій розвитку інтелектуальних систем керування ресурсами інфокомунікаційних сервісів за умови енергозбереження. Досліджено методи забезпечення якості надання послуг і формування вхідного математичного опису здатної навчатися системи керування інфокомунікаційним сервісом. Дослідження за темою проекту виконувалися в рамках створеної авторами проекту так званої інформаційно-екстремальної інтелектуальної технології (ІЕІ-технологія) аналізу даних, яка ґрунтується на максимізації інформаційної спроможності системи в процесі її навчання. На основі розроблених узагальнених критеріїв ефективності запропоновано методи оптимізації параметрів функціонування здатних навчатися систем керування ресурсами розподіленого обчислювального середовища.

СИСТЕМА ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ, НАВЧАННЯ, ОПТИМІЗАЦІЯ, КЕРУВАННЯ, ІНФОКОМУНІКАЦІЙНА СИСТЕМА, ІНФОРМАЦІЙНИЙ КРИТЕРІЙ, ІНФОРМАЦІЙНО-ЕКСТРЕМАЛЬНА ІНТЕЛЕКТУАЛЬНА ТЕХНОЛОГІЯ

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ 5

ВСТУП 6

1 КЕРУВАННЯ РЕСУРСАМИ ТА ПЛАНУВАННЯ ЗАВДАНЬ В ІНФОКОМУНІКАЦІЙНІЙ СИСТЕМІ 8

1.1 Методи планування завдань та ресурсів 8
1.2 Енергозбереження в інфокомунікаційних системах 21
1.3 Прогнозування функціонального стану вузлів розподіленого обчислювального середовища 26

2 КЕРУВАННЯ ЯКІСТЮ ОБСЛУГОВУВАННЯ В ІНФОКОМУНІКАЦІЙНИХ СИСТЕМАХ 43

2.1 Принципи забезпечення якості обслуговування в інфокомунікаційній системі 43
2.2 Ідентифікація трафіку та анонімність в інформаційно-телекомунікаційній системі 63
2.3 Виявлення атак та керування безпекою в інфокомунікаційній системі 77

3 ІНФОРМАЦІЙНИЙ СИНТЕЗ СИСТЕМИ КЕРУВАННЯ ІНФОКОМУНІКАЦІЙНИМ СЕРВІСОМ 88

3.1 Узагальнений критерій ефективності здатної навчатися системи керування інформаційно-телекомунікаційним середовищем 88
3.2 Алгоритми машинного навчання системи керування інформаційно-телекомунікаційним середовищем 96
3.3 Алгоритми оптимізації параметрів функціонування системи керування інформаційно-телекомунікаційним середовищем 118
3.4 Інформаційно-екстремальний метод ідентифікації типу мережевого трафіку 128
3.5 Метод прогнозування рівнів сприйняття якості обслуговування в інфокомунікаційних системах 140

ВИСНОВКИ 153

ПЕРЕЛІК ПОСИЛАНЬ 155

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ІЕІ-технологія – інформаційно-екстремальна інтелектуальна технологія;

КФЕ – критерій функціональної ефективності;

ОПР – особа, що приймає рішення;

СКД – система контрольних допусків;

СППР – система підтримки прийняття рішень;

ФВА – функціонально-вартісний аналіз.

ЦОД – центри оброблення даних.

ELA – угода про рівень очікуваної якості послуг (Experience Level Agreements).

FSS – пошук косяком риб (Fish School Search, FSS).

IDS – системи виявлення атак (Intrusion Detection System).

KPI – ключовий показник продуктивності (key performance indicators).

KQI – ключовий показник якості (Key Quality Indicator).

MOS – середнє значення експертних оцінювань (Mean Opinion Score).

PSO – оптимізація роєм частинок (Particle Swarm Optimization).

SLA – угоди про рівень послуг (Service-level agreement).

SVM – метод опорних векторів (Support Vector Machine).

QoE – якість послуги, що сприймається користувачем (Quality of Experience).

QoS – якість обслуговування (Quality of Service).

ВСТУП

Глобалізація світової економіки та формування соціально-орієнтованого середовища вимагають значних обчислювальних і телекомунікаційних ресурсів, систем накопичення, оброблення, аналізу і збереження великих обсягів даних. З цією метою мережі провайдерів, корпорацій, комунальні та державні ІТ-системи інтегрувалися в єдине розподілене обчислювальне середовище, ІТ-інфраструктура та сервіси якого утворили інфокомунікаційні системи різного призначення. При цьому особливої актуальності набуває задача інформаційного аналізу і синтезу систем керування плануванням завдань та розподілом ресурсів в розподіленому обчислювальному середовищі з метою забезпечення енергозбереження при високій якості обслуговування користувачів інфокомунікаційних сервісів. Основним напрямком розв'язання цієї важливої науково-практичної задачі є застосування ідей і методів машинного навчання та розпізнавання образів. Але відомі методи інтелектуального аналізу даних характеризуються невисокою функціональною ефективністю машинного навчання через науково-методологічні ускладнення, пов'язані із довільними початковими умовами формування вхідних навчальних матриць, багато вимірністю словника ознак і алфавіту класів розпізнавання, суттєвим апріорним перетином класів розпізнавання, впливом неконтрольованих збурюючих факторів тощо.

Як один із перспективних напрямків інформаційного синтезу систем керування інфокомунікаційними сервісами є створена в Сумському державному університеті так звана інформаційно-екстремальна інтелектуальна технологія (ІЕІ-технологія) аналізу даних, в рамках якої виконувалися дослідження за темою звіту. Основна перевага методів, створених в рамках ІЄІ-технології, полягає в тому, що вони

ґрунтуються на максимізації в процесі машинного навчання інформаційної спроможності системи, яка визначає її функціональну ефективність.

Заключний звіт складається із вступу, трьох розділів, висновків і переліку посилань.

Перший розділ присвячено аналізу сучасного стану і тенденцій розвитку інтелектуальних систем керування ресурсами інфокомунікаційних сервісів за умови енергозбереження.

У другому розділі викладено аналіз методів забезпечення якості надання послуг в інфокомунікаційнитх системах. При цьому основна увага приділена формуванню словника ознак і алфавіту класів розпізнавання, які характеризують функціональні стани системи керування інфокомунікаційним сервісом. Крім того, досліджено проблему виявлення атак і викладено сучасні методи кібербезпеки.

Третій розділ присвячено розробленню узагальнених критеріїв ефективності, на основі яких запропоновано методи оптимізації параметрів функціонування здатних навчатися систем керування ресурсами розподіленого обчислювального середовища.

Результати наукових досліджень, одержаних виконавцями проекту, опубліковано в працях [1-33].

1 КЕРУВАННЯ РЕСУРСАМИ ТА ПЛАНУВАННЯ ЗАВДАНЬ В ІНФОКОМУНІКАЦІЙНІЙ СИСТЕМІ

1.1 Методи планування завдань та ресурсів

Ефективне керування IT-інфраструктурою систем розподілених обчислень полягає у врахуванні та узгодженні інтересів користувача та власника ресурсів. Важливу роль тут відіграє планувальник обчислювальних завдань, який приймає рішення про надання обчислювальному завданню доступних ресурсів на основі метаданих вхідного завдання, прийнятих політик обслуговування та інформації про стан ресурсів.

У загальному випадку на вхід системи планування надходять завдання, що складаються з набору задач (рис. 1.1).

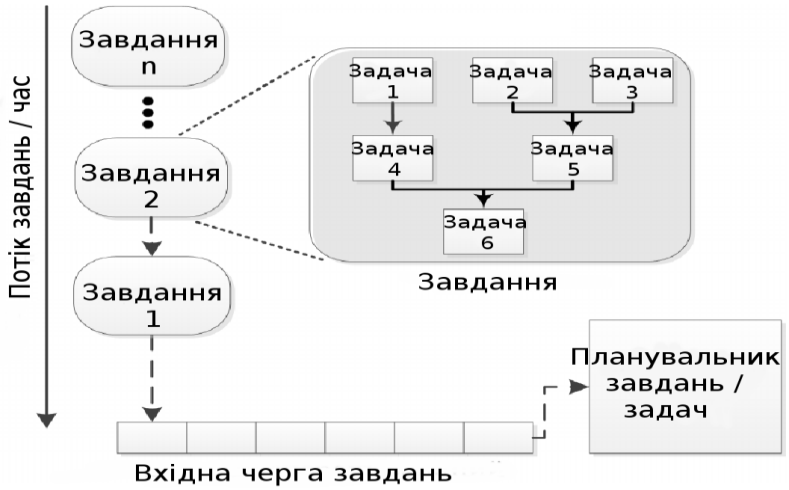


Рисунок 1.1 – Схема надходження завдань на вхід планувальника

Під обчислювальною задачею розуміють потік інструкцій для процесора з єдиним адресним простором, значеннями регістрів процесора, стеком, відкритими файлами, глобальними змінними тощо.

Потік задач надходить у вхідну чергу. Задачі є атомарною одиницею планування і в рамках одного завдання можуть бути незалежними чи організовані в дерево залежностей. Незалежні задачі виконуються паралельно, а при наявності залежностей – послідовно. Залежності визначають порядок виконання робіт і потоки даних між задачами. У випадку незалежних задач відмінності між плануванням завдань та задач відсутні. Планувальник крім виділення чи резервування ресурсів на вузлах обробки встановлює порядок, в якому задачі із вхідної черги будуть виконуватися на цих ресурсах.

Для розміщення обчислювальних завдань обираються прийнятні за ресурсом, часом та ціною слоти (рис. 1.2).

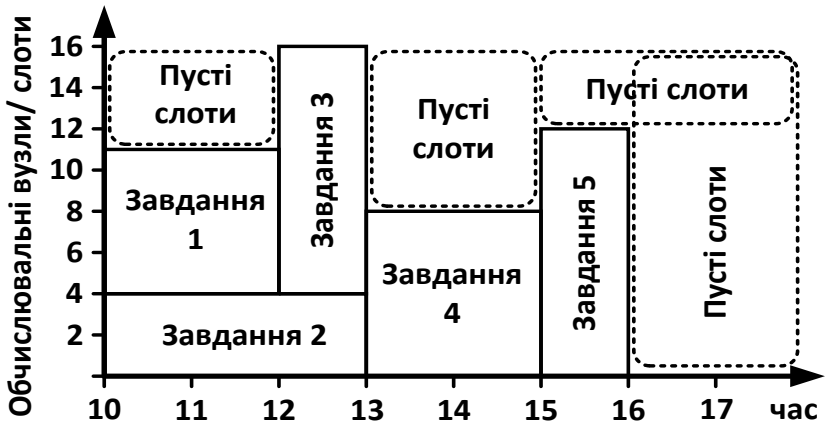


Рисунок 1.2 – Карта розподілу обчислювальних ресурсів для вхідних завдань

Для запуску будь-якого багатопроцесорного завдання здійснюється узгоджене виділення потрібних для її виконання слотів. У випадку однорідних обчислювальних вузлів сукупність слотів для виконання завдання представляє собою прямокутне вікно, а для процесорів з різною продуктивністю це вікно характеризується нерівним правим краєм

У загальному випадку ресурсами обчислювального середовища вважається сукупність програмних та апаратних засобів для виконання обчислювальних задач. Прикладом ресурсів є процесор, середовище

передачі даних, прикладне програмне забезпечення, система зберігання даних і тому подібне.

Процесори на обчислювальних вузлах розподіленого обчислювального середовища часто керують окремими ресурсами, які розділяються з декількома іншими процесорами чи пов'язані тільки з цими процесорами, тому доцільно розглядати узагальнене поняття віртуального вузла (рис. 1.3).

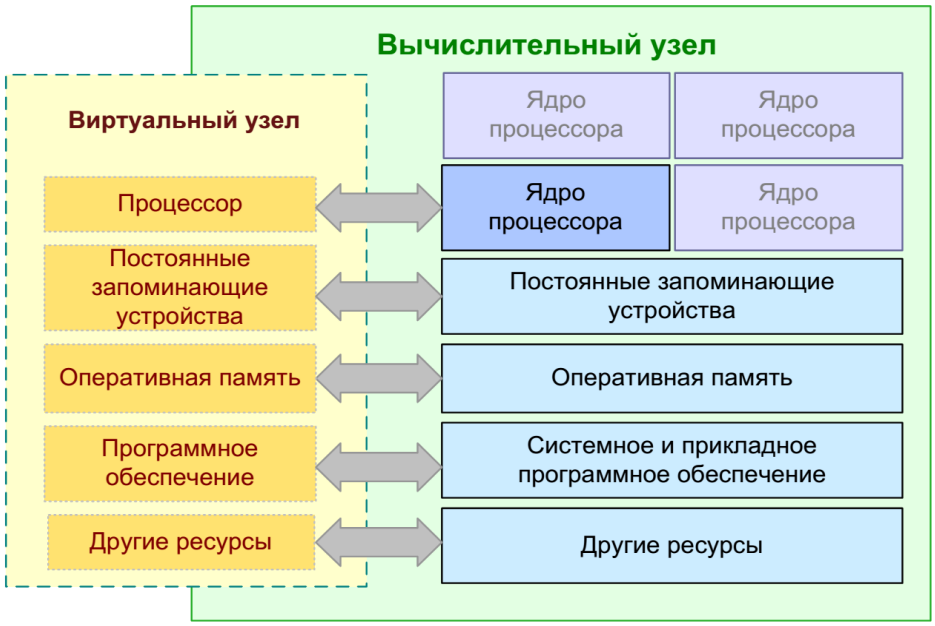


Рисунок 1.3 – Схема відношення ресурсів віртуального і фізичного вузлів розподіленої обчислювальної системи

Віртуальний вузол розглядається як один обчислювальний елемент (ядро процесора) у зв'язці із співставленими йому ресурсами. Якщо декілька процесорів використовують одні й ті ж ресурси, які характеризуються обсягом (оперативна пам'ять, постійна пам'ять і так далі), то часто вважають, що обсяг цих ресурсів ділиться рівномірно між усіма віртуальними вузлами, які відповідають даним процесорам. На практиці можлива конкуренція віртуальних вузлів за ресурси фізичного вузла, яку можна спрогнозувати і запобігти цій ситуації.

Алгоритми планування завдань в розподіленому обчислювальному середовищі можна поділити на статичні і динамічні.

Статичне планування і розрахунок вартісної оцінки обчислень здійснюється до початку виконання завдання, коли інформація щодо всіх ресурсів розподіленого обчислювального середовища і всіх завдань вже доступна. Одна з головних переваг статичної моделі – це простота реалізації планувальника. Однак, вартісна оцінка, заснована на статичній інформації, є не адаптивною до ситуацій, коли час відгуку стає більш тривалим ніж очікувалося через високе завантаження системи, або коли один з обчислювальних вузлів виходить з ладу, або стає ізольованим внаслідок мережевих відмов. Для вирішення проблеми використовують допоміжні механізми, такі як механізм перепланування.

Динамічне планування зазвичай застосовується, коли важко оцінити обчислювальну вартість додатків, що надходять на виконання динамічно в режимі online. Динамічне планування завдань включає в себе два важливих компоненти: оцінка стану системи і прийняття рішення про зв'язуванні завдання з черги з обраним ресурсом. Для збереження оптимального стану обчислювальної системи використовується балансування завантаження всіх її ресурсів. Перевага динамічного балансування завантаження над статичним полягає в тому, що система не зобов'язана знати про поведінку додатки під час його виконання до його запуску. Особливо це підхід корисний в системі, де основною метою є максимізація утилізації ресурсу, а не мінімізація часу виконання окремих завдань.

У динамічних сценаріях планування відповідальність за прийняття глобальних рішень планування ресурсів може лежати на одному централізованому планувальнику або декількох розподілених планувальниках. Централізована стратегія має перевагу, що полягає в простоті реалізації, але вона погано масштабується, невідмовостійка і часто стає вузьким місцем для продуктивності системи. У тому випадку,

коли вся інформація щодо стану ресурсів і завдань відома, оптимальна прив'язка завдань до ресурсів може бути зроблена на підставі деякої цільової функції, такої як мінімізація часу виконання завдань і максимальні утилізація ресурсів. Розподілені алгоритми планування можна розділити на пов'язані або непов'язані, залежно від того, як працюють вузли, що використовуються при плануванні завдань, спільно або незалежно (несумісно). У неспільному випадку локальні планувальники діють як автономні сутності і приймають рішення, з урахуванням їх власних цільових функцій. У спільному випадку кожен планувальник в розподіленому обчислювальному середовищі несе відповідальність за виконання його власної частини завдання планування, але при цьому всі планувальники працюють з однією спільною метою в масштабі всієї системи

При організації розподілених обчислень найпростішими є спискові алгоритми планування, які дозволяють обрати із списку вхідних завдань одне з них для призначення на вільні віртуальні вузли [46].

Спискові алгоритми планування впорядковують вхідні завдання в чергу за деяким їх параметром, наприклад, за кількістю необхідних для виконання процесорів або часом її виконання. А потім переглядають її, починаючи з голови, і призначають завдання на вільні обчислювальні ресурси.

Найбільш відомі спискові алгоритми планування: FCFS (First Come First Served) (першим прийшов, першим обслуговується), SJF (Shortest Job First) (найкоротша задача перша), LJF (Longest Job First) (найдовша задача перша), RJF (Random Job First) (випадкова задача обслуговується першою) і так далі.

Найбільш простою в реалізації є дисципліна FCFS (First Come First Served), відповідно до якої задачі обслуговуються «у порядку черги», тобто в порядку їх появи. Ті задачі, які були заблоковані в процесі роботи (потрапили в будь-який із станів очікування, наприклад, із-за операцій

вводу/виводу), після переходу в стан готовності стають в чергу готовності перед тими задачами, які ще не виконувалися. Іншими словами, утворюється дві черги: одна черга утворюється з нових задач, а друга черга – з тих, що раніше виконувалися, але потрапили в стан очікування. Такий підхід дозволяє реалізувати стратегію обслуговування «по можливості закінчувати обчислення в порядку їх появи». Ця дисципліна обслуговування не потребує зовнішнього втручання в хід обчислень, при ній не відбувається перерозподілу процесорного часу. До переваг цієї дисципліни, перш за все, можна віднести простоту реалізації та малі витрати системних ресурсів на формування черги задач, а також немає необхідності в інформації про тривалість виконання завдань. Однак поряд з простотою використання дисципліна FCFS призводить до того, що при збільшенні завантаження обчислювальної системи зростає і середній час очікування обслуговування, причому короткі завдання (потребують невеликих витрат машинного часу) змушені очікувати стільки ж, скільки і трудомісткі завдання. Уникнути цього недоліку дозволяють дисципліни SJN та SRT.

Дисципліна обслуговування SJF (Shortest Job First), що означає: наступним буде виконуватися найкоротше завдання, вимагає, щоб для кожного завдання була відома оцінка в потребах машинного часу. Необхідність повідомляти планувальнику характеристики завдань, в яких описувалися б потреби в ресурсах обчислювальної системи, привела до того, що були розроблені відповідні мовні засоби. Зокрема, мова JCL (Job Control Language) був одним з найбільш відомих мовних засобів. Користувачі змушені були вказувати передбачуваний час виконання, і для того, щоб вони не зловживали можливістю вказати свідомо менший час виконання (з метою отримати результати раніше за інших), ввели підрахунок реальних потреб. Диспетчер завдань порівнював замовлений час і час виконання і у випадку перевищення зазначеної оцінки в даному ресурсі ставив дане завдання не в початок, а в кінець черги. Дисципліна

обслуговування SJF припускає, що є тільки одна черга завдань, готових до виконання нею. І завдання, які в процесі свого виконання були тимчасово заблоковані (наприклад, очікували завершення операцій введення / виведення), знову потрапляють в кінець черги готових до виконання нарівні із новими вхідними завданнями. Це призводить до того, що завдання, яким потрібно дуже небагато часу для свого завершення, вимушені очікувати у черзі нарівні з тривалими роботами, що не завжди добре.

Для усунення цього недоліку була запропонована дисципліна SRT (Shortest Remaining Time, наступне завдання вимагає найменше часу для свого завершення).

Всі ці дисципліни обслуговування можуть використовуватися для пакетних режимів обробки, коли користувач не змушений очікувати реакції системи, а просто віддає своє завдання і через кілька годин отримує свої результати обчислень. Для інтерактивних обчислень бажано, насамперед, забезпечити прийнятний час реакції системи і рівність в обслуговуванні.

Досить популярний алгоритм, де віртуальні вузли утворюють віртуальне кільце з маркером, а наступна готова до виконання робота призначається на вузол, що має маркер, потім маркер передається наступному вузлу (Round Robin, RR, кругова). Дисципліна кругової диспетчеризації найбільше підходить для випадку, коли всі завдання мають однакові права на використання ресурсів центрального процесора. Однак одні завдання завжди потрібно вирішувати в першу чергу, тоді як інші можуть почекати. Це можна реалізувати за рахунок того, що одному завданню диспетчер задач привласнює один пріоритет, а іншій задачі – інший. Завдання в черзі будуть розташовуватися відповідно до їх пріоритетів.

В даний час для планування обчислень в масштабованих системах активно використовуються такі алгоритми, як міграція процесів і робіт,

спільне, пов'язане планування (Gang-scheduling) і алгоритм бек-філінг (Backfilling) або зворотного заповнення [47, 48].

Алгоритм зворотного заповнення (Backfilling) вимагає інформацію про тривалість виконання завдань. Мета алгоритму – найбільш щільно заповнити пусті вікна. Для цього серед наявних вікон обирається найбільш широке вікно, тобто з максимальною кількістю процесорів, і наступні роботи, які потрапляють в чергу, будуть призначатися на процесори цього вікна. Якщо нова робота не поміщається в жодне з доступних вікон, то вона поміщається в кінець черги. Таким чином, роботи поширюються у зворотний бік щодо шкали часу. Алгоритм зворотного заповнення досить часто використовується в системах черг, що здійснюють «справедливий» доступ користувальницьких задач до ресурсів багатопроцесорних систем. Переваги алгоритму зворотного заповнення такі:

1) дозволяє складати розклад для гетерогенних розподілених обчислювальних систем;

2) уникає зависання низько пріоритетних завдань в чергах, гарантуючи їх запуск;

3) може складати досить щільні за часом розклади;

4) має прийнятні характеристики швидкості роботи.

Алгоритм планування групами робіт (Gang-schedu-ling) здійснює розподіл ресурсів багатопроцесорної системи між групами робіт. Роботи об'єднані в групи за пріоритетами. Роботи однієї групи поділяють безліч процесорів таким же чином, як і у випадку алгоритму Backfill, однак допускається переривання робіт, якщо на багатопроцесорну систему надходить група робіт з великим пріоритетом.

Задача складання розкладу належать до класу *NP*-повних задач з ростом складності рішення щодо розмірності. Існуючі точні методи вирішення в гіршому випадку під час пошуку здійснюють перебір всіх можливих варіантів розподілу робіт за виконавцями, що вимагає великих

обчислювальних витрат для задач високої розмірності. Тому знаходження оптимального рішення задачі розподілу за прийнятний час стає важко досяжним.

Вирішення подібного класу завдань вимагає великих обчислювальних і, відповідно, часових ресурсів для знаходження оптимального рішення задачі, внаслідок чого виграш від використання знайденого рішення не покриває величезні витрати на його отримання. На практиці для вирішення *NP*-складних завдань часто використовують евристичні методи, що не гарантують знаходження оптимального рішення, але дозволяють досить швидко отримувати рішення прийнятної якості.

Евристичні методи оптимізації засновані на використанні різних розумних, в більшості випадків заснованих на життєвих та природних аналогіях, звичках, правилах, спрямованих на досягнення компромісу між прагненням до найкращого результату і до скорочення часу на перебір варіантів дій для досягнення цього результату. Незважаючи на недостатню теоретичну обґрунтованість, ці методи дозволяють отримувати прийнятні рішення при порівняно невеликих витратах часу та інших ресурсів. До переваг евристичних методів можна також віднести зручність їх реалізації на електронно-обчислювальних машинах навіть при вирішенні завдань високої розмірності. Недоліки даних методів полягають у складності оцінки їх фактичної ефективності, тобто близькості отриманих рішень до оптимальних. Крім того, для кожного евристичного підходу існують завдання, для яких застосування даного підходу або неможливе, або призводить до відверто поганих результатів. Це вимагає ретельного експериментального дослідження евристичних методів з метою виділення класів завдань, при вирішенні яких ці методи найбільш ефективні.

До найбільш ефективних і популярних евристичних методів відносяться так звані метаевристичні – узагальнені стратегії пошуку

оптимуму в просторі рішень. Як приклад можна навести алгоритм імітації відпалу (Simulated Anneal), жадібний алгоритм (greedy algorithm), генетичні (Genetic Algorithms) та мурашині алгоритми (Ant colony optimization) [49].

Алгоритм імітації відпалу реалізує упорядкований випадковий пошук на основі аналогії з процесом утворення в речовині кристалічної структури з мінімальною енергією при охолодженні. На практиці метод симуляції відпалу показує гарні результати, однак настройка параметрів алгоритму для вирішення конкретного завдання може забрати значний час. Ефективність методу істотно залежить від обраної схеми, зокрема, від початкової температури і функції охолодження.

Жадібний алгоритм полягає у прийнятті локально оптимальних рішень на кожному етапі, допускаючи, що кінцеве рішення також виявиться оптимальним. Процедура починається з вершини, що має найменший степінь. Маркуються всі сусідні вершини, а потім вершини, сусідні з сусідами. Перші n/p промаркованих вершин присвоюються одній підобласті, і процедура застосовується до решти графа до тих пір, доки всі вершини не є промаркованими.

Генетичні алгоритми (genetic algorithm) основані на використанні механізмів природної еволюції. Еволюція, за Дарвіном, здійснюється в результаті взаємодії трьох основних факторів: мінливості, спадковості та природного відбору. Мінливість є основою для утворення нових ознак і особливостей в будові і функціях організму. Спадковість закріплює ці ознаки. Природній відбір усуває організми, що погано пристосовані до умов існування. Цей алгоритм реалізує розвиток популяції хромосом (найчастіше у вигляді бітових рядків), де кожна хромосома кодує деяке рішення задачі в деякій її конкретній постановці. Така еволюція реалізується шляхом застосування декількох операторів, які імітують феномени живої природи (розмноження, мутація та інші). При реалізації планування завдань за допомогою генетичних алгоритмів масиви значень,

які визначають розклад, кодуються одним рядком, який в термінах генетичного алгоритму є хромосомою. Множина таких рядків утворюють популяцію можливих розкладів виконання завдань в розподіленому обчислювальному середовищі. Хромосоми оцінюються за допомогою фітнес-функції (функції придатності), яка характеризує якість рішення. В процесі селекції з урахуванням значень фітнес-функції ймовірнісним шляхом відбираються ті розклади, які будуть брати участь у формуванні наступного покоління. Генетичні оператори схрещування та мутації супроводжуються перевіркою на коректність новоутворених розкладів. Описаний процес повторюється доки зміни значення фітнес-функції від покоління до покоління стає незначним. В цьому випадку обирається розклад з максимальним значенням фітнес-функції. Проте при вирішення задач високої розмірності генетичні алгоритм демонструють наступні недоліки:

1) відносно висока обчислювальна трудомісткість;

2) відносно невисока ефективність на останніх фазах моделювання еволюції (оператори пошуку в еволюційних алгоритмах не орієнтовані на швидке потрапляння в локальний оптимум);

3) досі не вирішені проблеми самоадаптації.

Мурашині алгоритми основані на імітації самоорганізації соціальних комах шляхом використання динамічних механізмів, за допомогою яких система досягає глобальної цілі в результаті локальної низькорівневої взаємодії елементів. Колонія мурах може розглядатися як багатоагентна система, в якій кожен агент (мураха) функціонує автономно за простими правилами. Основу "соціальної" поведінки мурах складає самоорганізація. Принциповою особливістю такої взаємодії є використання елементами системи тільки локальної інформації, при цьому виключено будь-яке централізоване керування. Самоорганізація досягається шляхом взаємодії наступних чотирьох компонентів:

1) випадковість;

2) багатократність;

3) позитивний зворотній зв’язок;

4) від’ємний зворотній зв’язок.

Взаємодія відбувається через спеціальну хімічну речовину – феромон, що виділяється мурахами на пройденому шляху. Чим вища концентрація феромону на шляху, тим більше мурах буде по ньому рухатися. З часом феромон випаровується, що дозволяє мурахам адаптувати свою поведінку під зміни зовнішнього середовища. Застосування мурашиних алгоритмів має високу ефективність при розв’язані багатьох комбінаторних задач, зокрема при календарному плануванні. При цьому ефективність мурашиних алгоритмів зростає при збільшенні розмірності задачі оптимізації.

У більшості розглянутих методів передбачалося, що ресурси є однорідними. Зокрема, в більшості випадків робиться припущення, що всі ресурси мають однакову продуктивність та ціну. В цьому випадку задача вибору ресурсів значно спрощується, оскільки існує взаємозамінність ресурсів. Проте в наш час в практичних умовах ресурси все частіше є гетерогенними і відрізняються один від одного архітектурою, продуктивністю, пам’яттю, ціною, пропускною смугою і т.д. Вибір підходящого ресурсу для завдання користувача стає багатопараметричною задачею, причому окремі параметри є взаємозалежними. Відповідно ускладнюється і задача організації обчислювального процесу з урахуванням як інтересів користувачів, так і власників ресурсу.

При вирішенні оптимізаційної задачі планування, як правило, використовуються «системно-центричні» (system-centric) критерії, якы не враховують вимоги споживачів до якості обслуговування. Подібні методи планування розміщують завдання у відповідності до системних параметрів, які стосуються завантаження ресурсів або пропускної здатності системи. Планувальники приділяють увагу або мінімізації часу

відгуку (response time), сумарного часу очікування (waiting time) і фактичного часу виконання завдання (turnaround time), або максимізації загального завантаження ресурсів (resource utilization). У цій ситуації метою планування виявляється поліпшення становища системи в цілому, але не вимоги, пропоновані окремими споживачами.

Традиційні підходи до управління ресурсами орієнтовані на критерії, що відносяться до всієї системи, такі як використання інфраструктури і/або пропускна здатність. У той же час потреби хмарних інфраструктур вимагають розвитку таких моделей, які більшою мірою відображали б інтереси користувачів. Іншими словами, потрібно створення таких алгоритмів розподілу завдань, які забезпечували б максимум корисності для індивідуального споживача.

В силу принципово егоїстичної поведінки користувачів розподіленого обчислювального середовища їх важко змусити об'єктивно враховувати загальносистемні критерії. Тому паралельно розвивається використання стимулюючих економічних механізмів для організації управління ресурсами. Наприклад, ціна обчислень встановлюється відповідно до сумарної кількості заявок попиту і пропозиції. Часові відхилення від директивних строків виконання завдань можуть штрафуватися подорожчанням процесів проведення обчислень в обчислювальній системі. Ринково-вартісні міркування є важливою складовою сучасних досліджень в галузі розподілу обчислювальних ресурсів.

Слід зазначити, що застосування відомих алгоритмів організації обчислювального процесу в традиційних розподілених середовищах є малоефективним у хмарних системах у зв’язку з такими факторами:

1) випадкова зміна кількості користувачів, наслідком чого є непередбачувана динаміка попиту/ пропозиції/ доступності ресурсів, що вимагає динамічної масштабованості ресурсів;

2) зміна кількості користувачів і структури попиту викликає необхідність внесення коректив в плани обчислень;

3) багатофакторність, тобто наявність безлічі різних критеріїв, політик, переваг та обмежень на проведення обчислювальних робіт, що призводить до необхідності балансування між ними;

4) різноманітність вимог і переваг користувачів обчислювальної системи вимагає індивідуального підходу до споживачів сервісів;

5) із-за неточності знань про характеристики фізичних вузлів, віртуальних машин та задач відбувається розбалансування навантаження в ході виконання завдань, тобто система ресурсів перестає відповідати загальній стратегії планування за кількістю, якістю обслуговування та продуктивністю.

Всі ці фактори визначають особливості постановки задачі організації обчислювальних процесів в розподілених хмарних середовищах з динамічною структурою.

Таким чином, сучасні тенденції в організації розподілених обчислень вимагають динамічної масштабованості, динамічного балансування навантаження в умовах неточності знань про характеристики фізичних і віртуальних машин, поведінку та ресурсні вимоги завдань обробки з урахуванням з урахуванням інтересів користувачів та власників ресурсів.

1.2 Енергозбереження в інфокомунікаційних системах

В останні роки все більшої актуальності набуває питання підвищення енергоефективності обчислювальних систем. Споживання електроенергії становить основну частину експлуатаційних витрат в розподілених обчислювальних системах. Наприклад, обсяг споживання електроенергії в центрах обробки даних, що підтримують масові хмарні сервіси, може перевищувати обсяг споживання електроенергії цілих міст.

Існуючі праці в цьому напрямі можна поділити на два основні шляхи підвищення енергоефективності обчислювальних систем:

1) архітектура з низьким енергоспоживанням (статичний підхід);

2) зниження енергоспоживання програмними засобами (динамічний підхід).

Статичний підхід передбачає побудову обчислювальної системи з компонентів, що характеризуються низьким енергоспоживанням. Архітектурний підхід характеризується дороговизною, оскільки в його основі лежить застосування нестандартних компонентів та інженерних рішень. Однак останнім часом в цьому напрямі сталися значні зрушення, всі виробники обладнання намагаються знижувати його енергоспоживання, а конкуренція призводить до зниження вартості таких рішень.

Паралельно зі статичним підходом для зниження енергоспоживання розвиваються підходи на основі використання програмних засобів. Даний підхід об'єднує ряд методів, що дозволяють досягти енергоефективного виконання потоку завдань за рахунок вибіркового відключення або зниження продуктивності компонентів обчислювальної системи, коли вони простоюють або завантажені частково. Можна виділити три основні способи програмного підвищення енергоефективності:

1) відключення компонентів обчислювальної системи, що простоюють;

2) перерозподіл обчислювальних завдань за часом за умови наявності багатотарифної схеми оплати електроенергії (наприклад, день-ніч);

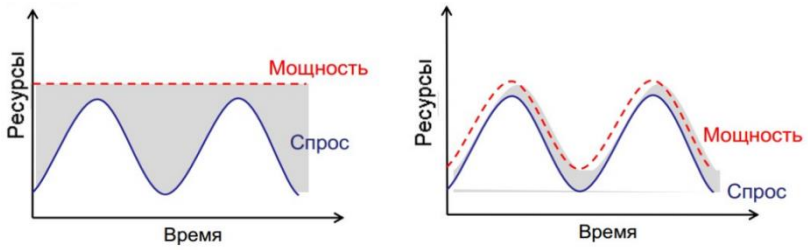
3) програмне керування продуктивністю компонент обчислювальної системи.

В нічний період, коли, як правило, тариф на електроенергію нижчий, варто збільшувати обчислювальне навантаження, а в день знижувати і компоненти, що простоюють, відключати. Загальна кількість спожитої енергії звичайно не зміниться, проте зменшиться її вартість, що також розглядається як підвищення енергоефективності.

Сучасні процесори і оперативна пам'ять мають можливість динамічно змінювати свою частоту і робочу напругу. Такий механізм носить назву DVS (dynamic voltage and frequency scaling). Основний принцип даного механізму полягає в тому, що при зниженні напруги процесора час обчислень збільшується, проте загальна кількість енергії витраченої на обчислення зменшується.

Важливий аспект динамічного керування енергоспоживанням полягає в тому, що зміна стану системи (підключення, зміна продуктивності тощо) має вартість *W*, що виражена в додатковій кількості спожитої енергії, затримці чи втраті продуктивності, що, взагалі кажучи, не гарантує зниження енерговитрат при переведенні системи в сплячий режим при відсутності роботи і назад в міру потреби. Наприклад, якщо вартість підключення W і завдання надходять у систему послідовно, кожна наступна відразу після завершення попередньої, то вимкнення системи після кожного завдання призведе до додаткової витрати енергії. Це робить завдання динамічного керування енергоспоживанням нетривіальним.

Технологія віртуалізації, що лежить в основі концепції хмарних обчислень, забезпечує гнучкість надання ресурсів і економію, що пов’язано з мінімізацією невикористаних ресурсів (рис. 1.4) [46].



а *б*

Рисунок 1.4 – Використання ресурсів хмари:

а – простої; *б* – оптимальне використання

За рахунок віртуалізації можна суттєво зменшити кількість серверів і необхідну площу дата-центра, а також знизити вимоги до потужності і охолодження в ньому. Хоча загальне енергоспоживання центру обробки даних знижується внаслідок впровадження віртуалізації, проте окремо взятий сервер потребуватиме більшої потужності. Так само при зменшенні загальної кількості серверів в дата-центрі кожна стійка (шасі) потребуватиме більше енергії, ніж раніше. При цьому експерти відзначають, що енергетична і охолоджуюча інфраструктури центрів обробки даних можуть стати неадекватними, коли структура споживання ресурсів повністю змінюється.

Як відомо, віртуалізація дозволяє по запиту розгортати, переміщувати чи клонувати додатки з однієї платформи на іншу навіть в процесі функціонування (рис. 1.5). При цьому потреби в енергозбереженні і охолодженні також можуть переміщуватися по площі дата-центру. Однак існуюча інфраструктура часто не в змозі забезпечити ці переміщення, не порушуючи обмеження, вказані в SLA [50, 51].

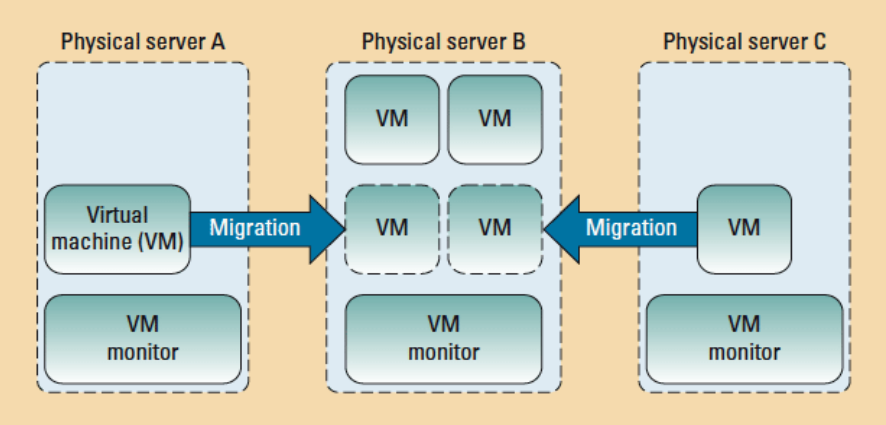


Рисунок 1.5 – Міграція віртуальних машин
на інший фізичний сервер

Площа датацентру завдяки віртуалізації зменшиться, але його загальна ефективність може залишитися субоптимальною, оскільки центр

обробки даних за рахунок віртуалізації оптимізований тільки частково. Підвищення щільності розміщення віртуальних машин в умовах гетерогенності додатків та конкуренції за спільні ресурси фізичних серверів може порушити умови SLA. Для комплексної оптимізації енерговитрат розподіленого середовища центрів обробки даних потрібно вирішити ряд важливих задач:

1) отримання моделі для прогнозування ситуації порушення умов SLA при різних комбінація розміщення віртуальних машин на фізичних серверах;

2) визначення моментів часу для переміщення віртуальної машини з перевантаженого сервера з метою уникнення зниження продуктивності і для переміщення віртуальної машини з недовантаженого сервера з метою підвищення ефективності використання ресурсів і зведення до мінімуму споживання електроенергії;

3) визначення оптимальної підмножини віртуальних машин, які підлягатимуть перерозподілу на інші сервери, з точки зору системних критеріїв ефективності та SLA;

4) визначення оптимального місця розміщення нової віртуальної машини, чи машин обраних для міграції, з точки зору системних критеріїв ефективності та SLA;

5) визначити коли і які фізичні сервери будуть вимкнені для енергозбереження і навпаки вчасно увімкнені для уникнення порушень SLA.

При вирішенні цих задач потрібно враховувати, що міграція віртуальної машини викликає комунікаційні та інші накладі витрати. Наприклад, міграція між серверами однієї стійки (шасі) може призводити до 100 мс простою, а міграція між центрами обробки даних може займати більше хвилини. Для завдань обробки даних, що направляються на віртуальні машини, потрібно дотримуватися правила: дані повинні бути розміщені ближче до процесу їх обробки. Тому алгоритми динамічного

планування завдань та керування ресурсами повинні враховувати рівень локальності даних (Data Locality). Прийнято виділяти чотири рівні локальності даних: локальність на вузлі (node locality), локальність на сервері (server locality), локальність на серверних стійках (rack locality) та локальність поза серверною стійкою (off rack locality). Чим більша локальність під час міграції віртуальних машин чи завантаженні даних на обробку тим менше простоїв у системі і відповідно менше нераціональних витрат електроенергії.

Таким чином, основним способом енергозбереження в сучасних розподілених обчислювальних середовищах є використання технології віртуалізації та алгоритмів перерозподілу навантаження між фізичними серверами для мінімізації їх простою. При цьому існує ризик порушення умов SLA, тому для реалізація цього способу необхідно додактково використовувати методи прогнозування функціонального стану фізичних серверів та віртуальних машин.

1.3 Прогнозування функціонального стану вузлів розподіленого обчислювального середовища

Розроблення ефективних алгоритмів розподілу ресурсів в обчислювальному середовищі ускладнено відсутністю або неточністю інформації про ресурсні вимоги та поведінку конкретного завдання на навантажених гетерогенних вузлах. Апріорна невизначеність функціонального стану вузла при виконанні завдання і неспроможність точної оцінки часу його виконання можуть призвести до виділення надлишкових ресурсів, які будуть простоювати, знижуючи завантаженість обчислювального середовища, або виділення недостатнього обсягу ресурсів, що призводить до накладних витрат, пов'язаних з процесом введення до експлуатації додаткових ресурсів чи міграцією задач на інший вузол.

Для отримання або уточнення знань про поведінку додатків на вузлах обробки, як правило, використовують дані трасування їх роботи. У загальному випадку ці дані можуть бути представлені у вигляді часових рядів споживання ресурсів, де k-та точка ряду містить відмітки математичного очікування $\mu[k]$ та середньоквадратичного відхилення $\sigma[k]$ (рис. 1.6) [52].

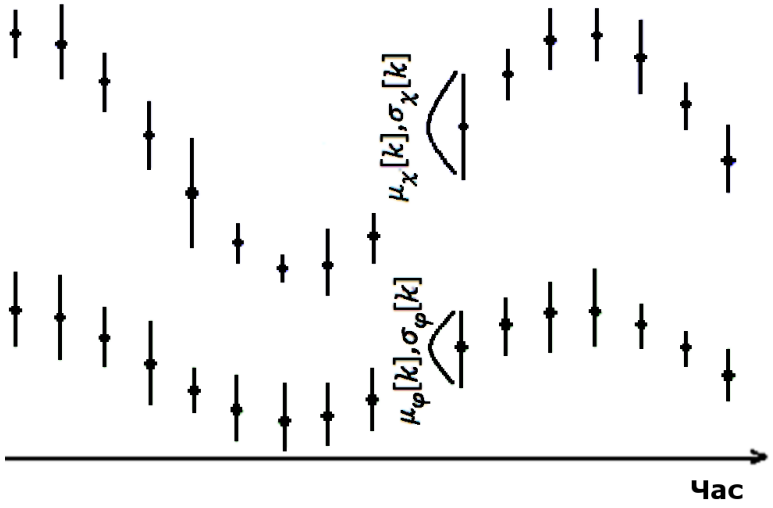


Рисунок 1.6 – Два часові ряди споживання обчислювального ресурсу із заданим в кожній точці середньоквадратичним відхиленням

З метою передбачення поведінки нового чи невідомого додатку в розподіленому обчислювальному середовищі система керування розподілом ресурсів може здійснити порівняння часових рядів споживання ресурсів відповідними задачами додатку з наперед відомими еталонами, які зберігаються в базі профілів. В базі профілів містяться еталонні зразки рядів споживання ресурсів для задач, що часто виконуються в середовищі. При цьому для кожного з профілів задано оптимізовані значення конфігураційних параметрів. Співпадання поведінки нового додатку з відомим шаблоном споживання ресурсів дозволяє реалізувати автоматичне підстроювання розподіленого

обчислювального середовища до вимог завдань, що покращує результати динамічного розподілу ресурсів і зменшує ймовірність порушення SLA.

Для формування еталонних зразків здійснюється профілювання відомих додатків з різними вхідними даними і настройками (рис. 1.7).

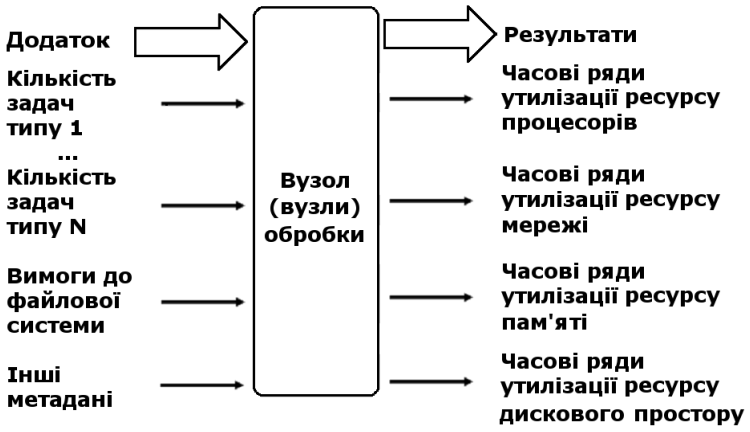


Рисунок 1.7 – Модель для профілювання додатків розподіленого обчислювального середовища

Відібрані завдання виконуються на різних обчислювальних вузлах декілька десятків разів. Утиліти моніторингу виконують трасування виконання завдань та обчислення статистичних характеристик споживання ресурсів в різні моменти часу від початку запуску.

Для обчислення схожості вхідного часового ряду з еталонним зразком, які мають різну довжину, часто використовується алгоритм динамічної трансформації шкали часу (Dynamic Time Warping, DTW). В цьому алгоритмі нерівності довжин долають повторним форму-ванням підвибірки з одного ряду перед порівнянням з іншим рядом. Для визначення схожості між двома часовими рядами споживання обчислювального ресурсу $X = [x_1, x_2, ... x_N]$ і $Y = [y_1, y_2, ... y_M]$ за умови $N \geq M$ DTW-алгоритм використовує наступні рекурсивні формули:

$$D(i,j)=\min\begin{cases}D(i,j-1);\\ D(i-1,j)+d(x_i,y_j);\\ D(i-1,j-1),\end{cases} \qquad (1.3.1)$$

$$d(x_i,y_j)=\left\|CPU(x_i)-CPU(y_j)\right\|, \qquad (1.3.2)$$

де $d(x_i,y_j)$ – евклідова відстань між відповідними точками в обох часових рядах; $CPU(x_i)$ – значення споживання обчислювального ресурсу в точці x_i ряду X.

Результатом обчислень за формулою (1.3.1) є матриця $D(X,Y)$, кожний елемент якої $D(i,j)$ відображає мінімальну відстань між $[X(x_1),Y(y_1)]$ і $[X(x_i),Y(y_j)]$. Тому $D[N,M]$ відповідає мірі схожості рядів X і Y. За умови $N \geq M$ для порівняння використовується ряд Y', який формується шляхом ітераційного повторення вибірки елементів з ряду Y і береться з довжиною, рівною довжині ряду X. Після пошуку мінімальної відстані між двома рядами за алгоритмом DTW буде сформовано новий ряд Y'. Остаточне вимірювання схожості між рядами Y' та X може бути здійснене шляхом обчислення відповідного кореляційного коефіцієнту:

$$CORR(X,Y')=\frac{1}{N}\sum_{i=1}^{N}(x_i-\mu_X)(Y'_i-\mu_{Y'}) \qquad (1.3.3)$$

Кореляційний коефіцієнт (1.3.3) показує на скільки схожі два часові ряди. Якщо $CORR(X,Y')=0$, то схожість між рядами відсутня, а у випадку $CORR(X,Y')=1$ робиться висновок про точне співпадіння. Для практичних задач, як правило, обирається емпіричний поріг $CORR(X,Y')\geq 0{,}9$, перевищення якого приймається за співпадіння рядів.

В розподіленій обчислювальній системі внаслідок реалізації масштабованої і гетерогенної архітектури збільшується її складність і ймовірність перевантажень та відмов в обслуговуванні. Найбільш загальною платформою для реалізації великомасштабних обчислювальних систем є кластери, в тому числі віртуальні. З метою підвищення відмовостійкості обчислювальної системи та доступності її сервісів використовуються алгоритми прогнозування перевантажень та відмов при запуску завдань користувача. Керуючі вузли кластерних систем повинні мати систему контролю доступу (Admission Control) з метою визначення сумісності завдань користувача з поточними можливостями гетерогенного обчислювального середовища. Система контролю доступу відхиляє рішення щодо призначення задач користувача конкретному вузлу, якщо існує висока ймовірність порушення конфігураційних обмежень, заданих в обчислювальному середовищі.

Конфігураційні обмеження обов’язково містять задані адміністратором ресурсні обмеження вузлів та задані або обрані користувачем вимоги до умов обслуговування. Ресурсні обмеження характеризують ємність мережевих, процесорних і дискових ресурсів вузла та граничний обсяг частки ресурсів, яка може виділятися задачам користувача залежно від його ролі (адміністратор, привілейований користувач, авторизований користувач, гість). Користувач у свою чергу визначає бюджет, призначений для оплати тарифікованих ресурсів, граничний час обробки задач (deadline), та заданий користувачем пріоритет “важливості”.

У праці [53] запропоновано евристичний алгоритм для контролю доступу до ресурсів кластера на основі моделі косинусної подібності. Суть алгоритму полягає у порівнянні вектора доступних на вузлі ресурсів з вектором потрібних задачі для її виконання ресурсів. При цьому як міра подібності використовується косинус кута між векторами.

Для з’ясування загального обсягу зайнятих ресурсів в обчислювальному середовищі складається вектор задач, що виконуються на i-му вузлі

$$T_{compound}(i) = T_1 + T_2 + \ldots + T_k + \ldots + T_n,$$

де T_k – вектор ресурсних вимог кожної із задач.

Для кожної задачі, що перебуває в черзі для виконання, так само складається вектор її ресурсних вимог

$$T_q = E_{cpu}e_1 + E_{mem}e_2 + E_{disk}e_3 + E_{nw}e_4,$$

де e_1, e_2, e_3, e_4 – базисні вектори;

E_{cpu} – середній обсяг процесорного ресурсу;

E_{mem} – середній обсяг оперативної пам’яті;

E_{disk} – середній обсяг дискового простору;

E_{nw} – середній обсяг мережевого ресурсу.

Вектор доступних ресурсів обчислюється як різниця між загальним обсягом ресурсів R_{total} та компонентами вектора $T_{compound}$, тобто

$$T_{availability} = R_{total} - T_{compound}.$$

У випадку призначення q-ї задачі на один з вузлів обробки обсяг невикористаних ресурсів можна обчислити за формулою

$$T_{unused} = T_{availability} - T_q.$$

Завдання, що виконуються на вузлах обробки, не повинні конфліктувати та конкурувати за спільні ресурси, тому для перевірки сумісності обчислюють косинусну міру подібності вектора вільних ресурсів до вектора ресурсних вимог за формулою

$$similarity = \frac{T_q \cdot T_{availability}(i)}{\left\|T_q\right\| \cdot \left\|T_{availability}(i)\right\|} = cos\ (\overrightarrow{T_q} \wedge \overrightarrow{T_{availability}(i)})$$

Косинусна міра відповідає косинусу кута між вектором T_q і $T_{availability}(i)$ і її значення лежить в діапазоні [-1; 1]. Чим більше значення міри подібності тим більша сумісність задачі з доступними ресурсами. При цьому більші значення косинусної міри відповідають меншим значенням невикористаних ресурсів. Тому при прийнятті рішення щодо розміщення задачі на конкретному вузлі обробки міру подібності потрібно об’єднати з оцінкою обсягу невикористаних ресурсів T_{unused} у вигляді нерівності

$$C_{heu} \le \alpha \cdot similarity + (1-\alpha) \cdot \left|T_{unused}\right|,$$

де C_{heu}, α – параметри конфігурації, що задаються адміністратором розподіленого обчислювального середовища.

У випадку гетерогенних ресурсів та обчислювальних задач визначення порогових значень параметрів, що характеризують місткість вузлів обробки, є складним завданням, яке адміністратор не може вирішити без використання інструментів статистичного або інтелектуального аналізу (Data Mining) даних. Тому одним з найефективніших підходів до прогнозування стану перевищення ресурсних обмежень перед призначенням задачі на обчислювальний вузол є побудова вирішальних правил на основі машинного навчання за

даними передісторії, де роль вчителя виконує прапорець перевищення ресурсних обмежень. При цьому вектор ознак, що розпізнається здатним навчатися класифікатором, складається з ресурсних ознак задачі, що знаходиться в черзі, та ресурсних ознак вузла, на який передбачається переміщення задачі (рис. 1.8).

Ресурсні ознаки задачі формуються на основі унікального ідентифікатора задачі та передісторії її обробки на заданому вузлі. До ресурсних ознак задачі відносяться середнє вибіркове значення та вибіркова дисперсія обсягу використаних ресурсів процесора, оперативної пам’яті, пропускної здатності мережі та пропускної здатності для операцій вводу-виводу при роботі з жорстким диском. Додатковими ознаками задачі можуть бути метадані вхідних блоків даних, які корелюють з ресурсними вимогами.

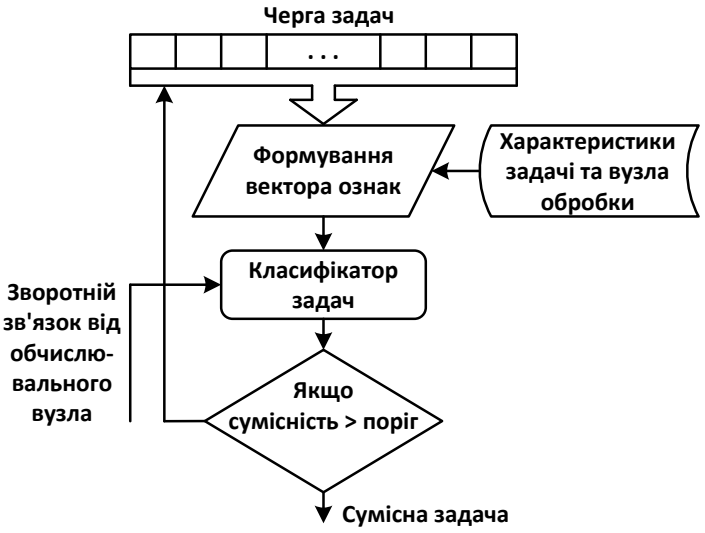


Рисунок 1.8 – Схема алгоритму вибору задач для їх розміщення на вільному вузлі обробки

Ресурсні ознаки вузла поділяють на статичні, які не змінюються протягом функціонування обчислювального вузла, та динамічні, які змінюються в процесі його функціонування. До статичних характеристик вузла відносяться кількість процесорів, частота процесорів, архітектура процесорів, повний обсяг фізичної оперативної та віртуальної (файл

підкачки) пам’яті, обсяг кеш пам’яті, обсяг пам’яті жорстких дисків, тип та версія операційної системи. До динамічних характеристик відносяться кількість використаних на вузлі процесорів, рівень завантаження процесорів, обсяг вільної оперативної пам’яті, інтенсивність операцій вводу/виводу, пропускна здатність мережі для прийому та передачі інформації, обсяг вільної віртуальної пам’яті, залишок дискового простору, споживана потужність, мережева відстань між розміщенням задачі та розміщенням блоків даних тощо. Якщо планувальнику не вдається знайти для задачі вузол, який буде задовольняти ресурсним обмеженням, то задача буде очікувати на появу необхідних ресурсів або буде повернута користувачу для корекції вимог до умов обслуговування.

Задача оптимального розміщення віртуальних машин на фізичних серверах полягає в одночасній оптимізації енергоспоживання та рівня утилізації ресурсів підключених серверів. На рис. 1.9 показано схему керування хмарною інфраструктурою, зосередженою на забезпеченні заданого рівня SLA [54].

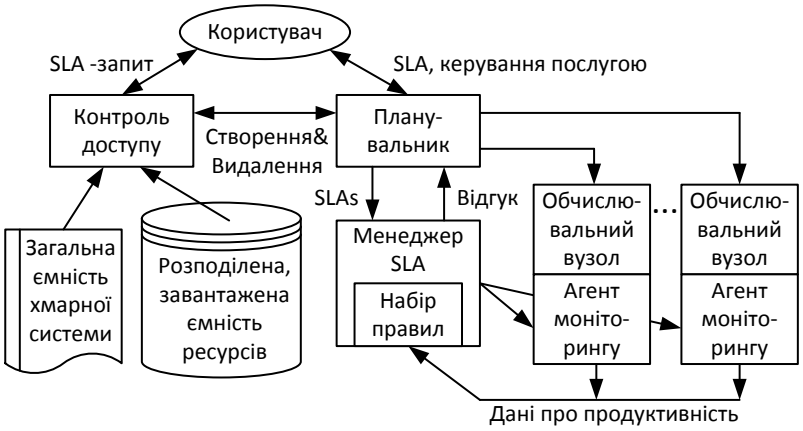


Рисунок 1.9 – Схема системи керування хмарною інфраструктурою

Агенти моніторингу (рис. 1.9) надають інформацію планувальнику про поточний стан фізичних та віртуальних ресурсів. Менеджер SLA на основі набору правил відкидає рішення планувальника, які призводять до деградації продуктивності в системі.

Для зменшення простою ресурсів, як правило, використовується динамічна оптимізація розміщення віртуальних машин, в тому числі з реалізацією оверселінгу (Over-selling), що полягає у перепродажі незадіяних зарезервованих ресурсів. При цьому система керування IT-інфраструктурою повинна вирішувати задачі енергозбереження шляхом розвантаження і вимкнення живлення слабозавантажених фізичних машин та попередження зниження продуктивності віртуальних машин, що розміщені на хостовому сервері з високим рівнем утилізації ресурсів. Основною причиною зниження продуктивності віртуальних машин, які працюють на спільній інфраструктурі апаратного забезпечення, є їх недостатня ізоляція. Ефект змагання віртуальних машин за різні фізичні ресурси ще називають “інтерференцією”. Інтереференція виникає на рівні апаратних компонентів, таких як процесори, пам'ять, засоби вводу-виводу та мережевий канал, і підсилюється їх спільним впливом. Прогнозування зниження продуктивності віртуальних машин дозволяє підтримувати заданий рівень обслуговування SLA шляхом врахування відповідної інформації при міграції існуючих чи розміщенні нових віртуальних машин на фізичних хостових серверах.

Обсяг різного типу ресурсів, який використовує будь-яка віртуальна машина, залежить від додатків, що працюють під її управлінням. Для спрощення віртуальні машини можуть бути класифіковані на такі типи:

1) віртуальні машини, що інтенсивно використовують ресурс процесорів;

2) віртуальні машини, що інтенсивно використовують ресурс оперативної пам’яті;

3) віртуальні машини, що інтенсивно використовують ресурс дискової пам’яті;

4) віртуальні машини, що інтенсивно використовують мережевий ресурс.

Для точної картини використання ресурсів віртуальною машиною необхідно розглядати вектор з чотирма компонентами, що відповідають рівню використання ресурсів кожного типу

$$R_{VM} =< E_{cpu}, E_{mem}, E_{disk}, E_{bw} >,$$

де E_{cpu} – середній відсоток використання процесора;

E_{mem} – середній відсоток використання оперативної пам'яті;

E_{disk} – середній відсоток використання дискової пам'яті;

E_{bw} – середній відсоток використання ресурсу мережі.

У випадку гетерогенного датацентру ресурсні вимоги віртуальної машини слід описувати вектором ресурсів, де кожен компонент розраховується як відсоток від загального обсягу ресурсів конкретного сервера, що є кандидатом на розміщення віртуальної машини. Наприклад рівень споживання процесорного ресурсу може бути розрахований за формулою

$$E_{cpu} = \frac{V_{cpu}^{VM}}{C_{cpu}^{PM}},$$

де V_{cpu}^{VM} – середній обсяг використання процесорного ресурсу віртуальною машиною;

C_{cpu}^{PM} – максимальна ємність процесорного ресурсу фізичного сервера.

Для оцінки можливостей фізичного сервера щодо розгортання додаткової віртуальної машини потрібно обчислити вектор сумарного використання ресурсів сервера всіма віртуальними машинами, що розміщені на ньому

$$R_{PM}^{used} =< E_{cpu}^{used}, E_{mem}^{used}, E_{disk}^{used}, E_{bw}^{used} >,$$

де $E_{cpu}^{used}, E_{mem}^{used}$, E_{disk}^{used}, E_{disk}^{used} – відсоток зайнятого ресурсу процесора, оперативної пам'яті, диску і мережі відповідно.

Аналогічно можна описати вектор незайнятих ресурсів фізичного сервера

$$R_{PM}^{free} =< E_{cpu}^{free}, E_{mem}^{free}, E_{disk}^{free}, E_{bw}^{free} >,$$

де $E_{cpu}^{free}, E_{mem}^{free}$, E_{disk}^{free}, E_{disk}^{free} – відсоток зайнятого ресурсу процесора, оперативної пам'яті, диску і мережі відповідно.

Якщо віртуальна машина щойно створена, то її шаблон споживання ресурсів апріорно невідомий, і може бути заданий за замовчуванням, наприклад у вигляді вектора з такими значеннями:

$$R_{VM} =< 25\%,25\%,25\%,25\% >$$

Для уникнення конкуренції віртуальних машин за доступ до ресурсів фізичного сервера варто розміщувати віртуальні машини за принципом несхожості їх ресурсних вимог, або за принципом схожості набору вільних ресурсів фізичного сервера з ресурсними вимогами віртуальної машини, що переміщується. Як критерій схожості (несхожості) можна використати косинусну міру, що обчислюється за однією з двох формул:

$$similarity = \frac{R_{VM} \cdot R_{PM}^{used}}{\left\| R_{VM} \right\| \cdot \left\| R_{PM}^{used} \right\|}; \qquad (1.3.1)$$

$$similarity = \frac{R_{VM} \cdot R_{PM}^{free}}{\left\| R_{VM} \right\| \cdot \left\| R_{PM}^{free} \right\|}. \qquad (1.3.2)$$

Пошук максимальної несхожості моделей споживання ресурсів за формулою (1.3.1) дозволяє підвищити консолідацію гетерогенних віртуальних машин. Пошук максимальної схожості за формулою (1.3.2) дозволяє підвищити раціональність використання ресурсів на фізичних машинах і одночасно зменшує ймовірність виникнення конкуренції віртуальних машин за спільні ресурси сервера.

У випадку обмежених ресурсів датацентру прогнозування рівня інтерференції дозволяє приймати рішення оптимальні у вартісному сенсі. При цьому повторюваний характер задач, що вирішуються додатками віртуальних машин, забезпечує можливість застосування методів машинного навчання для аналізу log-даних трасування роботи віртуальних машин і синтезу моделі прогнозування рівня їх інтерференції. На рис. 1.10 показано узагальнену схему системи прогнозування зниження продуктивності віртуальних машин внаслідок розміщення на сервері додаткової віртуальної машини. Вхідний математичний опис системи, показаної на рис. 1.10, може бути сформований за результатами кластер-аналізу даних трасування роботи віртуальних машин. Утворені таким чином групи (класи) віртуальних машин мають схожі середні значення споживання різного типу ресурсів фізичного сервера. При цьому словник ознак класифікатора віртуальних машин повинен включати середній обсяг використання ресурсу процесорів, оперативної пам’яті, файлу підкачки, мережевого каналу, дискового простору, середню інтенсивність операцій вводу-виводу з дисковою пам’яттю тощо. Так само в процесі кластер-аналізу може бути сформований алфавіт рівнів інтерференції, де словник ознак включає

записи про зміну у відсотках сумарного споживання різного типу ресурсів, зміну метрик продуктивності, відсоток помилок обробки задач на віртуальній машині, відсоток часу перебування процесора в стані блокування та інші зміни, викликані розміщенням нової віртуальної машини.

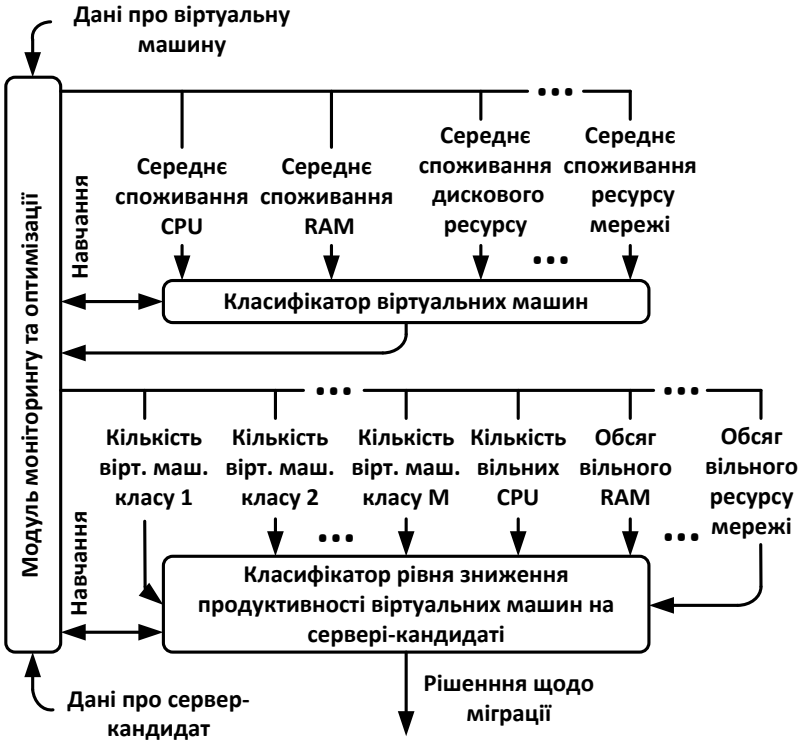


Рисунок 1.10 – Схема системи прогнозування рівня зниження продуктивності віртуальних машин

Останній крок аналізу даних передісторії полягає у зв’язуванні кожної знайденої комбінації розміщення віртуальних машин на фізичних серверах з відповідним класом рівня інтерференції. Як ознаки можна розглядати кількість розміщених на хості віртуальних машин кожного класу та метрики споживання ресурсів процесора, оперативної та дискової пам’яті і мережевого каналу.

В процесі машинного навчання здійснюється синтез класифікатора рівнів інтерференції віртуальних машин при різних комбінаціях їх розміщення на фізичних серверах. При цьому належність віртуальної машини до одного з шаблонів споживання ресурсів враховується при прогнозі рівня інтерференції.

Як бізнес-модель хмарні обчислення покликані надати користувачам високодоступне, надійне, масштабоване і недороге динамічне обчислювальне середовище. Проте, поширення хмарних обчислень а, отже, і створення великомасштабних центрів обробки даних по всьому світу, які містять величезну кількість взаємопов'язаних обчислювальних вузлів, призвели до нових проблем в управлінні цим обчислювальним середовищем: відмови та високе енергоспоживання. Відмови вузлів є характерними для великомасштабованих розподілених систем, які можуть мати тисячі вузлів, що виконують різноманітні завдання. За оцінками дослідників з Лос-Аламоської Національної Лабораторії (Los Alamos National Laboratory) середній час між відмовами вузла в петафлоп системи і становить 1,25 години.

Існує інформація про те, що випадки відмов стануть набагато поширенішими в найближчі десять років, що серйозно впливатиме на продуктивність та витрати на експлуатацію. Крім цього внаслідок поширення хмарних обчислень значно зростає обсяг споживання електричної енергії центрами обробки даних, що призводить до високих експлуатаційних витрат і викидів вуглекислого газу в навколишнє середовище. Ці фактори змушують постачальників хмарних сервісів шукати шляхи скорочення, пов’язаних з цим експлуатаційних витрат.

Відмови, що виникають в обчислювальному середовищі, швидко поширюються і чинять вплив на роботу сервісів та користувачів. При цьому виявлення збоїв та відмов часто займає значний час, оскільки апаратний рівень фізичних серверів прихований за шаром віртуалізації. Тому досі лишається актуальною задача оперативного виявлення

несправностей і реагування на них, перш ніж вони спричинять збитки. Один із перспективних підходів до оперативного виявлення несправностей полягає у прогнозуванні будь-якої відмови сервісів на основі аналізу журналів повідомлень, що містять інформацію про поведінку пристроїв обчислювальної системи.

В праці [54] запропоновано використовувати приховану напівмарківську модель для аналізу порядку повідомлень в журналі та виявлення послідовностей, пов’язаних з відмовами. Проте, застосування цього методу до великої системи, наприклад, хмарних центрів обробки даних, ускладнено рядом проблем:

1) формати повідомлень в гетерогенній системі різні;

2) порядок повідомлень в журналі може не відповідати порядку їх генерації в системі внаслідок затримок при зборі і записі;

3) старіння результатів навчання прогностичної системи внаслідок оновлення апаратного чи програмного забезпечення компонентів розподіленого обчислювального середовища.

Дослідниками з компанії Fujitsu було запропоновано знаходити статистичну залежність між множиною повідомлень у журналі подій та відмовами шляхом співставлення комбінації повідомлень, отриманих протягом заданого періоду, із прецедентами передаварійних повідомлень. При цьому ознаками передаварійного стану є кількість повідомлень заданого типу, отриманих протягом всього часового вікна спостереження (рис. 1.11).

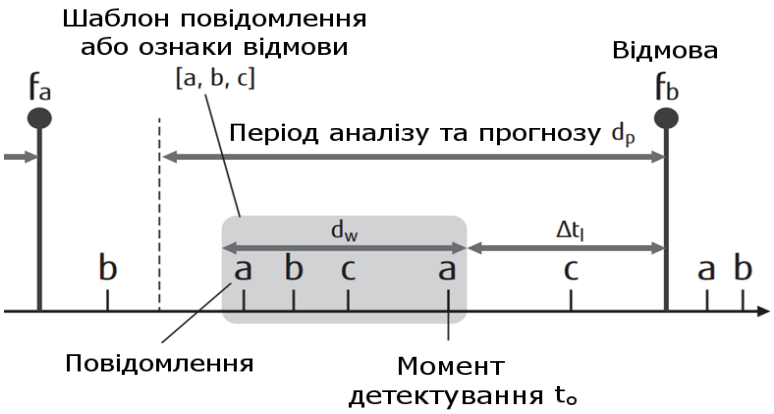


Рисунок 1.11 – Періоди аналізу повідомлень та прогнозування

Для обчислення кожної ознаки необхідно класифікувати тип повідомлення шляхом розбиття тексту кожного повідомлення на слова та обчислення статистики входження слів повідомлення до словника повідомлень.

З метою підтримки актуальності вирішальних правил всі виявлені відмови в сервісах повинні співставлятися з комбінацією повідомлень протягом часового вікна спостереження та оновлювати навчальну вибірку аварійних ситуацій. Так само і спрогнозовані відмови, що пройшли валідацію, використовуються для доповнення вибірки.

Таким чином, функціональний стан компонентів розподіленого обчислювального середовища може бути спрогнозований, що дозволяє зменшити ймовірність зниження продуктивності, перевантажень та відмов під час оптимізації енерговитрат.

2 КЕРУВАННЯ ЯКІСТЮ ОБСЛУГОВУВАННЯ В ІНФОКОМУНІКАЦІЙНИХ СИСТЕМАХ

2.1 Принципи забезпечення якості обслуговування в інфокомунікаційній системі

За умов стрімкого росту популярності веб-сервісів, технологій віртуалізації та розподілених хмарних обчислень телекомунікаційна інфраструктура виконує одну з найважливіших функцій, забезпечуючи інформаційний обмін між територіально розподіленими компонентами складних інфокомунікаційних систем. Активне впровадження концепції "Все як послуга" (Everything as a service, EaaS) і застосування сервіс-орієнтованої архітектури (Service Oriented Architecture, SOA) при побудові даних систем обумовлює підвищення уваги до проблем забезпечення необхідного рівня якості обслуговування. При цьому якість комплексних інфокомунікаційних послуг може бути забезпечена лише за умов укладання і строгого дотримання SLA всіма провайдерами/операторами, що приймають участь в наданні даних послуг.

Для задоволення вимог по доставці інформації в телекомунікаційних мережах розроблені методи, які іменуються методами забезпечення якості обслуговування (QoS) [55].

Методи забезпечення якості обслуговування в пакетних мережах можна класифікувати, відповідно до схеми, зображеної на рис. 2.1. Все різноманіття відомих методів забезпечення QoS можна поділити на дві основні групи. До першої групи слід віднести методи збільшення кількості ресурсів мережі. Дані методи передбачають використання засобів, що збільшують пропускну спроможність фізичних каналів, наприклад, за рахунок застосування високошвидкісного середовища

передачі або поліпшення характеристик сигналів.

Рисунок 2.1 – Класифікація методів забезпечення QoS в пакетних мережах

Питанням проектування телекомунікаційних мереж присвячено багато робіт, в яких обґрунтовується вибір обсягу мережевого ресурсу та його використання в процесі функціонування мережі. Але здійснивши обґрунтування вибору обсягу мережевих ресурсів, не завжди можна досягти максимізації показника ефективності їх використання. На практиці складно забезпечити високий рівень QoS в мережах тільки за рахунок збільшення їх фізичних ресурсів, оскільки необхідно вирішувати і інші не менш важливі завдання, пов'язані з її раціональним розподілом.

До основних засобів керування ресурсами традиційно відносять механізми керування чергами (буферный ресурс), засоби розподілу пропускної здатності трактів передачі (канальний ресурс), механізми керування трафіком (інформаційний ресурс).

Забезпечення наскрізної QoS "із кінця в кінець" (end-to-end) в рамках гетерогенної телекомунікаційної системи передбачає

використання цілого комплексу засобів керування (рис. 2.2).

Рисунок 2.2 – Базова архітектура керування мережевими ресурсами

Формування трафіку реалізується з метою зміни параметрів вхідного трафіку у відповідності з заданим профілем, вказаним в SLA. У випадку порушення параметрів профілю (зокрема, при перевищенні тривалості надсилання або середньої швидкості) здійснюється відкидання або маркування пакетів цього потоку (рис. 3)

Рисунок 2.3 – Керування вхідним трафіком на приграничному вузлі

Профілювання трафіку на основі правил політики (policing) передбачає у випадку порушення параметрів профілю (наприклад, перевищення тривалості пульсації або середньої швидкості) відкидання пакета чи його маркування зі зниженням пріоритету. Відкидання окремих пакетів знижує інтенсивність потоку і приводить його параметри у відповідність до вказаних в профілі. Маркування пакетів без відкидання необхідна для того, щоб пакети обслуговувалися, але вже зі зниженою якістю.

Функція вирівнювання трафіку (shaping) призначена для надання трафіку, що пройшов профілювання, необхідної «форми» в часі і реалізується шляхом буферизації пакетів. В основному за допомогою цієї функції намагаються згладити пульсації трафіку, і тим самим скоротити черги на вузлах мережі. Вирівнювання доцільно використовувати для відновлення часових співвідношень трафіку додатків, що працюють з рівномірними потоками, наприклад, додатків передачі голосу.

У випадках, коли підвищення навантажень або тимчасові коливання трафіку викликають перевантаження каналів зв'язку повне або часткове вирішення проблем продуктивності мережі може здійснюватися методами оптимізації маршрутизації. Задача оптимізації маршрутизації полягає в знаходженні такого рішення для маршрутизації трафіку на заданих структурі мережі і матриці попиту трафіку, яке призведе до оптимального QoS в мережі. Як міра QoS може розглядатися утилізація каналів, що пояснюється її впливом на затримку та втрату пакетів між маршрутизаторами. При цьому суть оптимізації полягає в "підстроюванні" маршрутизації до поточного навантаження з метою кращої утилізації мережевих ресурсів, що, в свою чергу, підвищує якість мережевих послуг QoS.

При маршрутизації, основаній на пункті призначення пакетів, маршрутизатор визначає вихідний інтерфейс для подальшого пересилання пакетів, виходячи зі значень метрик, які кількісно описують дистанцію до

пункту призначення. Як правило, окрема адитивна метрика присвоюється кожному каналу, потім алгоритм визначення найкоротшого шляху використовується для визначення оптимальних маршрутів між усіма вузлами мережі (однометрична маршрутизація). Тобто, задаючи відповідні значення метрик каналів, можна побічно впливати на схему маршрутизації і, таким чином, оптимізувати її.

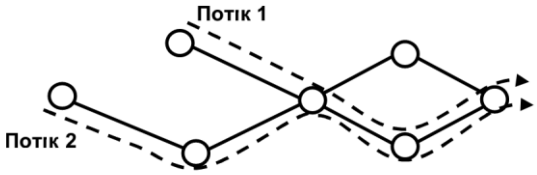
Прикладом багатометричної маршрутизації є протокол маршрутизації EIGRP, розроблений компанією Cisco, який враховує чотири метрики. Однак тільки дві з них використовуються за замовчуванням – адитивна метрика "затримка" і метрика "ємність".

Оптимізація маршрутизації здійснюється в процесі пошуку мінімуму максимального значення утилізації при різних значеннях метрики кожного з каналів. Критерій оптимізації має вигляд

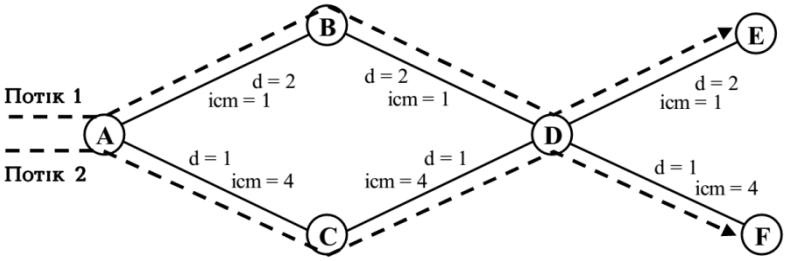
$$Критерій = \left(\frac{1}{\max\limits_{i \in канали} (утилізація_i)} \right)^p, \; p > 0.$$

Вектор параметрів в оптимізаційній задачі складається з вагових коефіцієнтів метрик, що є компонентами метрики маршруту, кожного з каналів. Процедура пошуку багаторазово застосовує можливі модифікації метрик для відведення трафіку із каналу з найвищою утилізацією. Процедура відведення трафіку повторюється доки вона покращує поточний результат.

На рис.2.4*а* показано, що при перетині двох потоків з однаковим пунктом призначення вони об’єднуються і далі надсилаються по одному й тому ж інтерфейсу, що може викликати перевантаження одних каналів і недовантаження інших.



а



б

Рисунок 2.4 – Сценарії маршрутизації при перетині двох потоків з однаковими пунктами призначення: *а* – без оптимізації; *б* – з оптимізацією

Аналіз рис. 2.4*б* показує, що маніпулюючи в процесі оптимізації адитивними метриками “затримка” та “ємність” можна відвести трафік від найбільш завантаженого каналу і здійснити довантаження альтернативних шляхів передачі.

Для пошуку оптимальних рішень можна використати будь-який з евристичних алгоритмів багатопараметричної оптимізації, які мають достатню оперативність, наприклад методи еволюційного програмування, такі як генетичний або рою частинок. У випадку використання генетичного алгоритму при однометричній маршрутизації кожен генний ланцюжок буде містити вагові коефіцієнти всіх каналів в порядку їх нумерації. У випадку двометричної маршрутизації генетичному алгоритму потрібно мати два генні ланцюжки для кожного рішення. Певне рішення виводиться з генного ланцюжка відповідним обчисленням. З метою збільшення швидкості обчислень перед оцінкою рішень можуть застосовуватися процедури локального пошуку.

Використання перелічених вище і їм подібних компонентів для конструювання маршрутних метрик, зазвичай реалізуються на евристичній основі без використання якого-небудь систематичного підходу. У зв'язку з цим, одним із перспективних напрямів рішення задач маршрутизації і визначення маршрутних метрик за умов апріорної невизначеності є використання апарату нечіткої логіки (Fuzzy Logic). Наприклад, питання віддавання переваги дистанції або завантаженості буфера в значній мірі суб’єктивне і неоднозначне, тому вибір рішення можна сформувати на основі нечіткої логіки, розглядаючи два параметри: дистанція та завантаженість пам'яті відповідного інтерфейсу (рис. 2.5).

Рисунок 2.5 – Модель маршрутизатора в рамках інтерфейсу

Якщо ввести лінгвістичні змінні Fuzzy_metric, distance та queue_length, які описують нечітку комбіновану та дистанційну метрики і коефіцієнт завантаженості вихідного буфера інтерфейсу відповідно, та задати кожній з них два терми Big та Little, то правила нечіткого висновку можна подати у вигляді:

якщо distance=Little і queue_length=Little то Fuzzy_metric=Little;

якщо distance=Big і queue_length=Little то Fuzzy_metric =Big;

якщо distance=Little і queue_length=Big то Fuzzy_metric= Big;

якщо distance=Big і queue_length=Big то Fuzzy_metric= Big.

На рис. 2.6. зображено поверхню вхід-вихід для нечіткого висновку, де “злам” поверхні визначає точку найбільшої невизначеності у виборі маршруту.

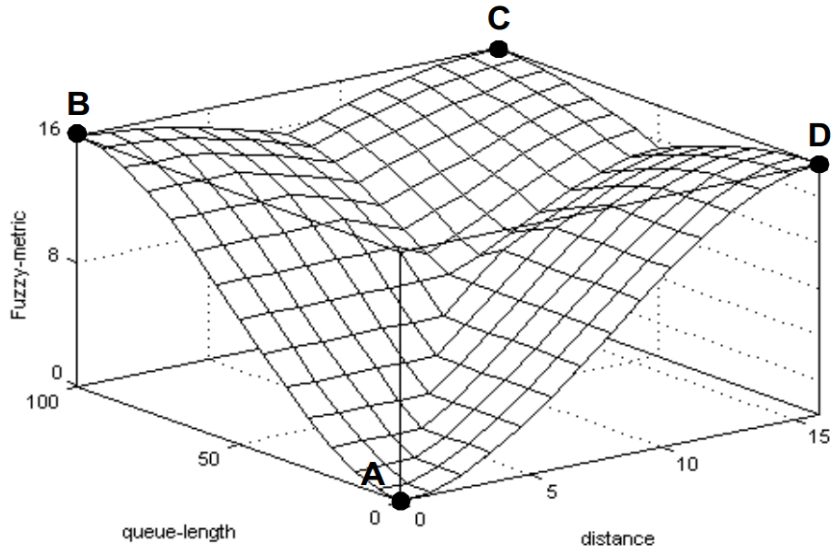


Рисунок 2.6 – Поверхня нечіткої метрики маршруту

Точка A на рис. 2.6 визначає позитивне тривіальне рішення про вибір маршруту, точки B, C і D – негативні тривіальні рішення (прийнятні маршрути відсутні).

Використання нечіткої логіки дозволяє створювати легкі в масштабувані системи маршрутизації, логіка роботи яких дає можливість розширювати і змінювати структуру метрики маршруту на систематичній основі шляхом зміни бази правил.

Нова технологія для маршрутизації MPLS (Multiprotocol Label Switching) дає можливість встановлювати структуру маршрутизації всередині IP-мережі незалежно від використовуваного протоколу

маршрутизації і задавати певні маршрути для окремих потоків трафіку. Кожен IP-пакет, призначений для MPLS маршрутизації, містить спеціальну мітку, яка лежить в основі прийняття рішень маршрутизаторами про подальше пересилання пакету на всьому шляху його проходження. Такий підхід дає високий рівень гнучкості і дозволяє досягти будь-якої бажаної картини маршрутизації. Однак дороговизна обладнання, проектування та обслуговування стримують широке впровадження MPLS. Крім того, для технології характерна складність в підтримці інформаційної безпеки – якщо не працює один протокол, то вся мережа не функціонує.

Основою ядра сучасних пакетних транспортних мереж є технологія IP/MPLS. здатна забезпечити параметри QoS. Встановлення логічних з'єднань виконується на базі роботи протоколів маршрутизації IP мережі – тобто спочатку за допомогою протоколів динамічної маршрутизації у вузлах мережі будуються таблиці маршрутизації, за допомогою іншого протоколу на базі даних таблиць маршрутизації автоматично прокладаються канали передавання інформації LSP (Label Switching Path) і будується MPLS мережа. Тому надійність, якість та керованість MPLS мережі значною мірою визначається якістю роботи протоколів динамічної маршрутизації.

Для керування мережевими компонентами (пристрої, лінії зв'язку, користувачі, додатки) з метою забезпечення надійності та QoS необхідно виконувати велику кількість задач, серед яких розгортання, конфігурування, моніторинг, налагодження, обробка помилок, ремонт і так далі. Дослідники в галузі комп'ютерних мереж давно визнали факт необхідності впровадження інтелектуальних компонентів для автоматизації обслуговування мереж, їх самостійної адаптації і вдосконалення в умовах постійно зростаючих розмірів, складності та гетерогенності.

Методи штучного інтелекту забезпечують інтелектуальний та

динамічний контроль і керування мережевими системами. Машинне навчання часто використовують для ідентифікації типу мережевого трафіку з метою застосування політик пріоритезації, виявлення вторгнень та виявлення несправностей мережевих пристроїв для своєчасного відновлення режиму нормального функціонування.

Кластер-аналіз успішно використовується для виявлення аномальної мережевої активності, що може становити загрозу як інформаційній безпеці, так і функціонуванню мережі взагалі. Прикладами вторгнень можуть бути Probe (сканування), DoS (відмова в обслуговуванні), R2L (проникнення), U2R (перевищення привілеїв) тощо. Прикладами реакції системи керування безпекою на мережеві вторгнення можуть бути розрив з'єднання з атакуючим об'єктом, блокування його на міжмережевому екрані, відслідковування шляху проникнення атакуючого об'єкта в систему, що захищається, і тому подібне.

Більшість відомих систем керування телекомунікаційними мережами використовують одну и ту ж базову структуру і способи взаємодії, основані на моделі взаємодії системи керування мережею NMS (Network Management System) з об'єктами керування за допомогою агентів (рис. 2.7).

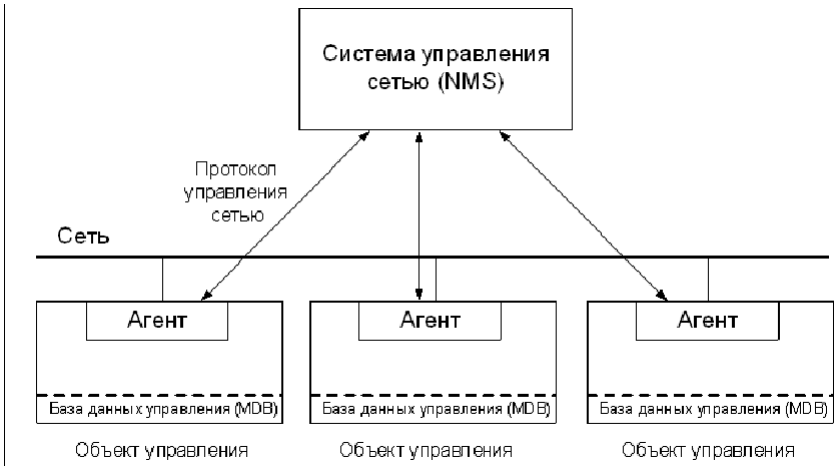


Рисунок 2.7 – Типова архітектура системи керування мережею

В комп’ютерних мережах системи мережевого керування використовують один із стандартних протоколів (SNMP чи CMIP). CMIP використовується в телекомунікаційних мережах, де необхідні всі доступні можливості керування мережами, в той час як SNMP використовується в локальних і корпоративних мережах, де достатньо мінімуму даних.

Забезпечення заданого рівня якості обслуговування в системах мережевого керування реалізують за допомогою так званих інтелектуальних агентів, тобто агентів здатних адаптуватися до змін стану мережі. При цьому для управління складною телекомунікаційною мережею доцільно використовувати розподілену систему керування (рис. 2.8), яка в максимальній мірі відображає структуру мережі.

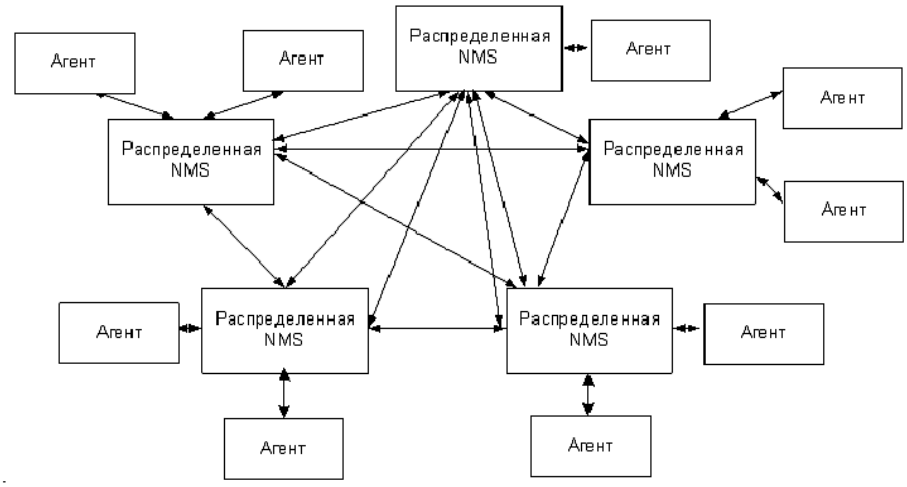


Рисунок 2.8 – Розподілена система керування телекомунікаційною мережею

Для динамічно реконфігурованих телекомунікаційних мереж необхідна більш гнучка, адаптивна система керування. Розв’язок цього завдання досягається шляхом побудови інтелектуальних систем керування телекомунікаційними мережами.

В задачі QoS входить забезпечення надійного, високошвидкісного,

неперервного зв’язку, що найскладніше гарантувати в мобільних мережах, оскільки топологія мережі змінюється при переміщенні вузлів. В цьому випадку велику роль в підтримці QoS і забезпеченні надійного зв’язку грає механізм хендовера (handover). Функція хендовера полягає у реалізації можливості переходу абонентів із зони дії однієї базової станції в зону дії іншої без втрати розмови чи передачі мультимедійного трафіку.

Мобільна мережа повинна мати базовий механізм, який забезпечує функціональні можливості: QoS, захист і мобільність. Користувачі прагнуть отримати сервіс, який задовольняє їх потреби за прийнятну ціну. Таким чином, існує необхідність в більш інтелектуальній мережі і в більш інтелектуальних сервісах.

На рис. 2.9 представлена узагальнена архітектура знання-орієнтованої мобільної мережі, що представляє собою деяку інтелектуальну надстройку над існуючою традиційною архітектурою.

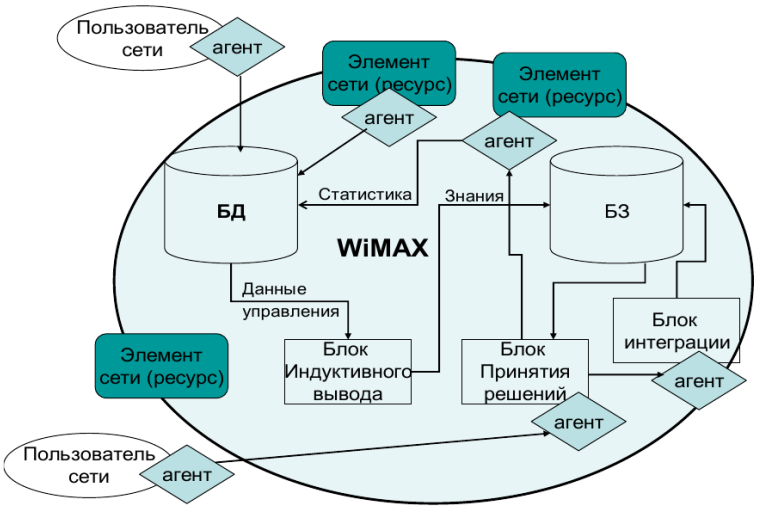


Рисунок 2.9 – Архітектура знання-орієнтованої мобільної мережі

Головною метою нової архітектури NGN є:

1) забезпечення якості і різноманітності сервісів для користувачів;

2) ефективне керування службами QoS;

3) забезпечення мобільності (хендовер і місце розташування

абонента) в мережі;

4) служба захисту.

Інтелектуальний вузол керування мережею працює на основі керуючої інформації, що надходить до вузлів, абонентів та інших компонентів мережі (використовуючи традиційні протоколи SNMP, CMIP) в базу даних MIB, де вона накопичується і зберігається для подальшої обробки.

Блок індуктивного висновку аналізує цю статистичну інформацію (деякі відхилення у функціонуванні мережі, що вимагають керуючого впливу) і одержує з даних певні закономірності – нові знання про правила функціонування мережі для того, щоб потім прийняти рішення про виконання певних керуючих впливів, які забезпечують безперебійне і якісне функціонування мережі.

Отримані закономірності, що стосуються конкретної ситуації в мережі, перетворюються і зберігаються в базі знань. Ці правила дозволяють використовувати накопичений в процесі функціонування мережі досвід (так, якщо раніше деякі рішення приводили до успішного результату, то в подальшому це ж рішення буде застосовуватися при схожих умовах без додаткового аналізу). Це дозволяє також прискорити реакцію системи на зміну зовнішнього середовища. Вирішальні правила з бази знань надходять до блоку прийняття рішень вузла, який формує сигнали керування для компонент мобільної мережі для корегування (покращення, оптимізації) її роботи.

Останнім часом з’явилась тенденція до концентрації ресурсів у центрах обробки даних і їх надання підприємствам і компаніям у вигляді сервісу на потребу, що сприяє розвитку IT-бізнесу та конвергенції сервісів і провайдерів. Сучасні оператори та провайдери зв’язку надають сервіси Інтернет, контенту, сервіси за моделями інфраструктура як сервіс IaaS (Infrastructure as a Service), платформа як сервіс PaaS (Platform as a

Service) та програмне забезпечення як послуга SaaS (Software as a Service).

Питання якості обслуговування QoS є критичним фактором для успіху хмарних послуг, оскільки від цього залежить репутація і прибуток оператора/провайдера. При цьому дотримання лише договору SLA не достатньо для забезпечення задоволеності користувачів хмарних сервісів. Функціональні параметри KPI (максимальний час відгуку, пропускна здатність, коефіцієнт помилок тощо) та інтегральні нефункціональні параметри KQI (наприклад, своєчасність, масштабованість, доступність тощо) мають контекстно-залежний сенс. Ці параметри мають різний вплив на якість для всіх осіб, залучених до процесу надання хмарних послуг. Для уникнення відтоку клієнтів існує потреба реалізації зворотного зв'язку для контролю сприйняття рівня якості послуг QoE і його врахування при балансуванні навантажень, що особливо актуально для послуг з високим пріоритетом (рис. 2.10) [56].

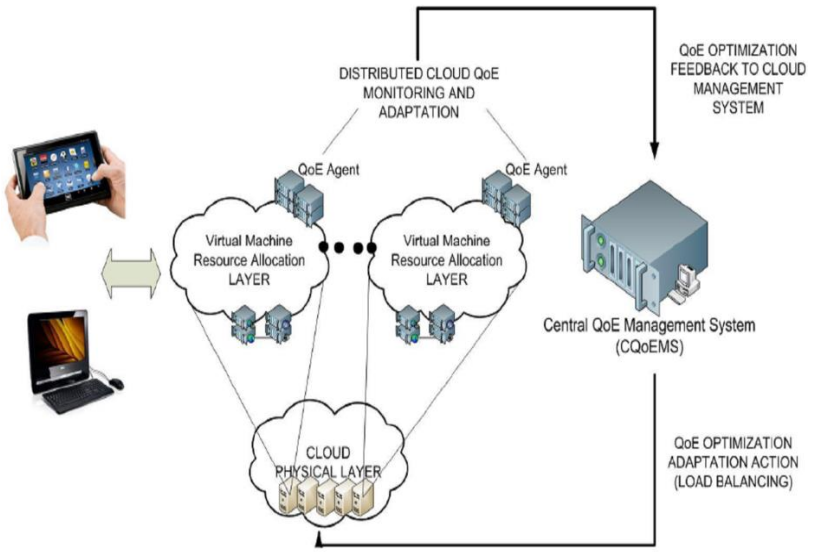


Рисунок 2.10 – Схема керування рівнем QoE в хмарному середовищі

QoE може постійно змінюватися, що потребує періодичного опитування QoE-агентів, що розміщуються на віртуальних машинах

хмарного середовища. Для прогнозування рівня QoE в режимі реального часу при виконанні оптимізаційних задач системою керування можна використовувати модель, що пов’язує рівень QoE з поточними показниками KPI та KQI і характеристиками цінової політики. Таку модель, як правило, отримують за допомогою методів машинного навчання. QoE може бути включена до SLA з метою визначення якості обслуговування з позицій кінцевого користувача.

Незважаючи на те, що керування хмарною інфраструктурою з урахуванням QoE орієнтоване на ринок, де рівень QoE розглядається провайдерами як потенційний прибуток, керування хмарним середовищем повинно здійснюватися економічно ефективно з урахуванням витрат на забезпечення якості обслуговування (рис. 2.11).

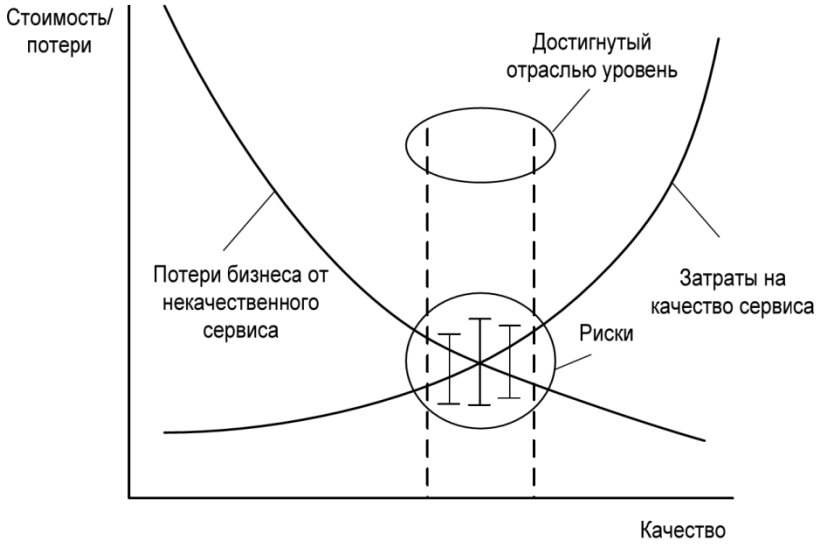


Рисунок 2.11 – Знаходження оптимального відношення рівня якості сервісу та вартості досягнення цієї якості

Вимога максимізації QoE-послуг і мінімізація пов’язаних з цим витрат C знаходяться в природній суперечності. Це призводить до необхідності встановлення економічно обґрунтованого обмеження рівня QoE для послуг з урахуванням як можливостей компанії, так і досягнутого IT-галуззю рівня та очікувань клієнтів. З метою впливу на

поточне значення QoE необхідно здійснювати оптимізацію метрик QoS як на рівні обчислювальних ресурсів, так і на мережевому та бізнес рівнях (рис. 2.12) [57].

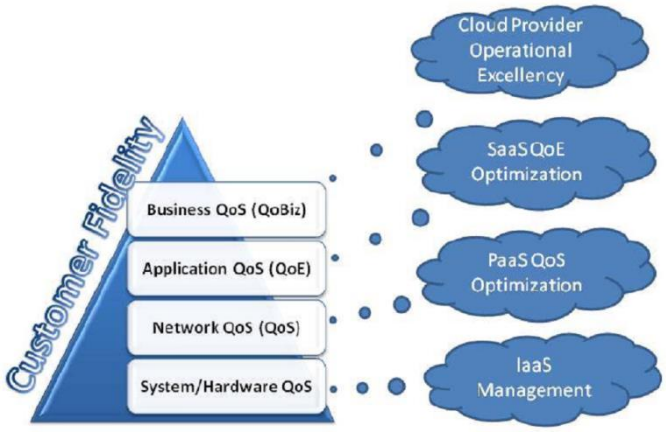


Рисунок 2.12 – Рівні керування в хмарному середовищі

Коли QoE-агенти повідомляють про зниження QoE система керування здійснює багатовимірний вплив для підтримки лояльності клієнтів та прибутковості хмар. Компонентами впливу можуть бути перегляд цінової політики, запит на збільшення чи перерозподіл мережевих та обчислювальних ресурсів.

Мережі провайдерів, корпорацій, комунальні IT-системи інтегрувалися, утворивши єдине IT-середовище. IT-інфраструктура, як його основа, перетворилися у важливий об’єкт керування. Сегмент ринку систем керування IT-інфраструктурою насичений продуктами, створеними на основі IT Infrastructure Library (ITIL) і концепції IT Service Management (ITSM). У процесній моделі ITIL виділяють три групи процесів управління: управління інфраструктурою (проектування і планування, розширення, супроводження і технічна підтримка); підтримка обслуговування (управління інцидентами і проблемами, конфігурацією і змінними, релізами); управління наданням послуг (рівнем обслуговування, фінансами, IT-послугами, готовністю,

неперервністю обслуговування, потужностями). Суттєвими недоліками відомих продуктів є висока вартість, використання закритих фірмових технологій, моделей і алгоритмів управління, відсутність засобів оцінки та керування QoE, засобів зміни оптимізаційних алгоритмів, критеріїв та обмежень.

У працях [58, 59] вітчизняних вчених розроблено і детально описано декомпозиційно-компенсаційний спосіб організації управління рівнем послуг в IT-середовищі, заснованого на декомпозиції задач управління і компенсації негативного впливу окремих чинників, таких як збільшення кількості користувачів, відмови в IT-інфраструктурі та інші, за рахунок виділення додаткових ресурсів критичним застосуванням. Для реалізації даного підходу розробники пропонують використовувати дворівневу модель системи керування з координатором (рис. 2.13).

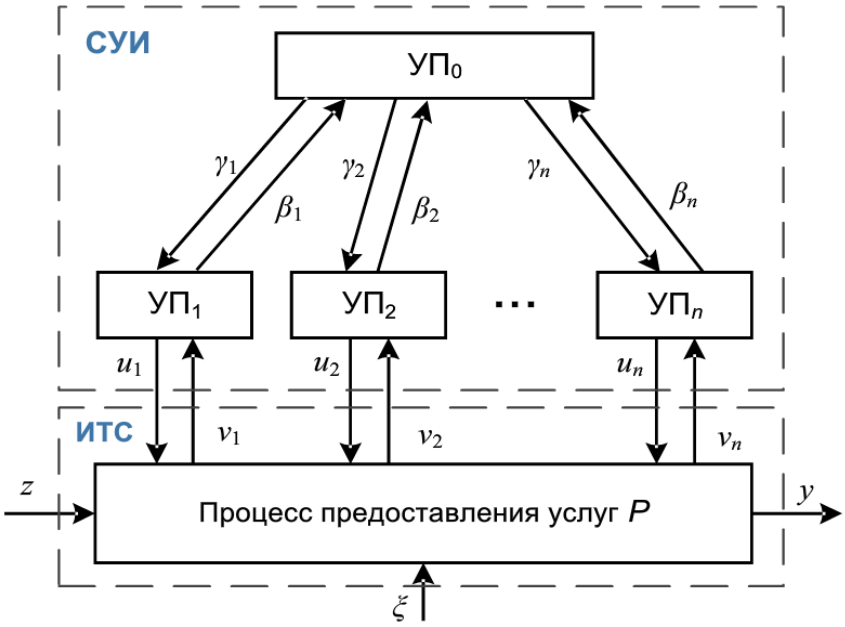


Рисунок 2.13 – Дворівневе представлення системи керування IT-інфраструктурою

Показаний на рис. 2.13 блок $УП_0$ виконує функції координатора, а розміщені нижче блоки $УП_1$,..,$УП_n$ є керуючими підситемами. Р –

керований процес, що протікає в ІТ-інфраструктурі. Координуючі сигнали $\gamma_1,..,\gamma_n$ впливають на підсистеми керування $УП_1,..,УП_n$, примушуючи їх діяти узгоджено підкоряючись єдиній політиці, орієнтованій на досягнення глобальної цілі, не дивлячись на те, що ця ціль може протирічити локальним цілям підсистем. Командні сигнали $u_1,..,u_n$ від $УП_1,..,УП_n$ до процесу Р є керуючими. Знизу вгору надходять сигнали зворотного зв'язку: від процесу Р до $УП_1,..,УП_n$ – $v_1,..,v_n$, і від керуючих підсистем до координатора $\beta_1,..,\beta_n$. При цьому на процес Р, що представляє собою керовану підсистему, крім керуючих сигналів $u_1,..,u_n$ надходять вхідні сигнали $z \in Z$ (запити користувачів) та сигнали збурюючого впливу $\xi \in \Xi$ (несправності в ІТ-інфраструктурі, функціональні відмови, запити інших користувачів, які є завадою для даних користувачів та інші впливи, що ускладнюють досягнення цілей керування).

Оскільки система керування функціонує в умовах невизначеності, неповноти і недостовірності інформації, наявності факторів ризику, множини конфліктуючих критеріїв і цілей підсистем системи керування, то від такої системи керування вимагається не досягнення оптимального функціонування ІТ-інфраструктури, що практично неможливо, а покращення якісних характеристик роботи ІТ-інфраструктури. Координатор узгоджує самостійні рішення і дії підсистем системи керування для покращення роботи ІТ-інфраструктури з точки зору якості надання ІТ-послуг. При цьому дії координатора направлені на покращення глобальної функції якості надання послуг, а прийняті ним рішення здійснюються в умовах невизначеності.

На початковому етапі даного підходу визначається обґрунтоване значення інтегрального рівня SLA на основі можливих втрат провайдера внаслідок незадовільного рівня обслуговування і його витрат на надання сервісів з визначеним рівнем із врахуванням ризиків. Решта етапів

пов’язані з декомпозицією проблеми визначення мінімальної вартості досягнення рівня обслуговування на підпроблеми компенсаційного визначення цієї вартості на підмережах, сервісах, ресурсах і технологіях. Задача системи керування полягає в протидії збуренню шляхом ітеративного керування за відхиленням. При цьому визначення сигналу похибки керування здійснюється на основі агрегації метрик, що вимірюються на рівні процесу до метрик, з якими оперує координатор.

Використання декомпозиційно-компенсаційного підходу дозволяє створювати ієрархію рішень в управлінні і підтримці узгодженого рівня послуг з урахуванням існуючих ресурсних обмежень і повноважень рівнів у виборі керування, задіюючи можливості верхніх рівнів ієрархії для вибору керування при неможливості реалізації керування на нижніх рівнях. Основними недоліками підходу можна вважати ігнорування індивідуальних користувацьких QoE при обгрунтуванні SLA та агрегації метрик нижчих рівнів, виключення з розгляду надання підсистемам керування прогностичних властивостей.

Оператори та провайдери інформаційно-телекомунікаційних послуг часто мають справу з надмірним резервуванням ресурсів, що призводить до марного споживання енергії та підвищення експлуатаційних витрат. Тому набула поширення практика оверселінгу (overselling), яка полягає в продажі ресурсів в більшому обсязі, ніж є в наявності. При цьому в основі оверселінгу лежить переконання, що більшість клієнтів ніколи повністю не використовують зарезервовані ресурси згідно тарифного плану, тому невикористані ресурси перепродаються іншим клієнтам. Одночасно з цим в технології хмарних обчислень розвивається принцип “оплата в міру використання” (pay-as-you-go), згідно якого в міру необхідності ресурси можна докупити або відмовитися і сплачувати лише за реально спожиті ресурси. Реалізація цих підходів потребує миттєвої еластичності ресурсів (миттєвого масштабування).

Згідно принципу еластичності ресурси, необхідні для реалізації послуги, виходячи з потреб користувача можуть швидко надаватися, розширюватися, стискатися і вивільнятися. Однак миттєве введення в експлуатацію нових ресурсів часто не можливе, оскільки необхідний деякий час для обробки запиту, перерозподілу ресурсів чи підключення фізичних ресурсів до мережі живлення і запуску програмного забезпечення. Такі затримки можуть викликати неприйнятно великий час відгуку для критично важливих додатків, до порушення SLA чи зниження QoE, і як наслідок до втрати доходів провайдера. Вирішити проблему можна шляхом прогнозування потреб додатків в ресурсах, що дозволяє системі керування здійснювати необхідний розподіл або введення до експлуатації додаткових ресурсів з деяким випередженням – до моменту реального запиту на задоволення потреб. Реалізацію прогнозування можна здійснити на основі технік обробки та ідентифікації сигналів для виявлення моделей короткострокового попиту ресурсів (сигнатури). Якщо не вдається вилучити сигнатури, то короткострокове прогнозування можна здійснити на основі методів розпізнавання образів. Результат прогнозування можна використати як для випередженого запиту на додаткові ресурси, так і для обмеження ресурсів для некритично важливих додатків до рівня, що не порушує SLA чи не знижує QoE. Зрозуміло, що реалізація прогностичних функцій не повинна призводити до невиправданих витрат, відповідні алгоритми повинні мати не високу обчислювальну складність.

Таким чином, у загальному випадку для реалізації ефективного керування інформаційно-телекомунікаційною системою передбачається розподіл інтелекту по всіх рівнях системи, надаючи їй властивості самоуправління та самоорганізації. Конфігурація та функціональність IT-інфра-структури повинні автоматично змінюватися залежно від вимог користувача. При цьому передбачається, що система не тільки реагує на поточні запити користувача, але також аналізує його вподобання і

поточне оточення, надаючи системі керування відповідну інформацію. Важливою задачею є скорочення часу відгуку інформаційно-телекомунікаційної системи, що можливо досягти при наданні системі керування здатності прогнозувати потреби та відгук користувачів (QoE).

2.2 Ідентифікація трафіку та анонімність в інформаційно-телекомунікаційній системі

Сучасні інформаційно-телекомунікаційні мережі характеризуються високими і надвисокими показниками пакетообороту, що обумовлює необхідність пріоритезації трафіку відповідно до вимог рівня обслуговування користувачів (SLA) та якості мережевих сервісів (QoS/QoE). Для визначення пріоритету при формуванні смуги пропускання окремого трафіку необхідно мати інструмент його точної ідентифікації (класифікації).

Методи ідентифікації трафіку, окрім забезпечення механізмів фільтрації та пріоритезації інформаційних потоків, можуть бути використані як зловмисниками, так і спеціальними державними службами для отримання конфіденційної інформації. Навіть при використанні захищених каналів (наприклад, SSL / TLS або SSH з'єднання) за допомогою класифікаційного аналізу пакетованого мережевого трафіку можна розпізнати різні види мережевої активності користувача, наприклад, відрізняти веб-навігацію, віддалене управління робочим столом, передачу файлів, пошук в Google в режимі захищених веб-сесій. Отримана інформація в деякій мірі може порушувати недоторканість приватного життя.

Класифікація мережевого трафіку активно досліджується протягом останніх років. Запропоновані методи класифікації трафіку, основані на аналізі портів призначення, корисного навантаження та характеристик потоку пакетів [60]. Всі ці методи мають свої переваги і недоліки,

обмеження в застосуванні (рис. 2.14).

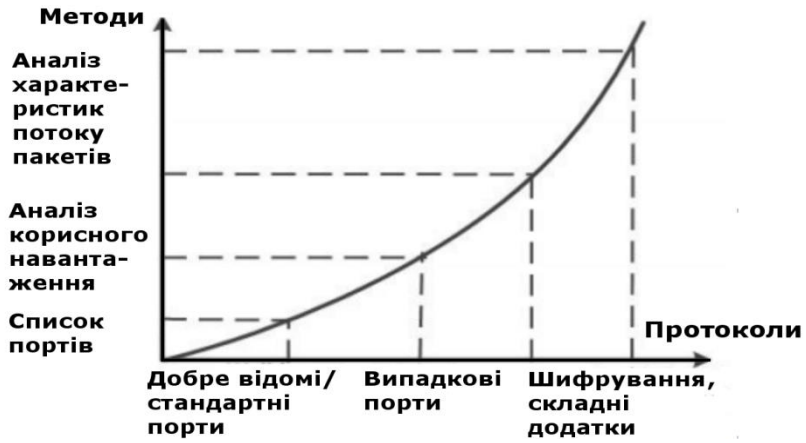


Рисунок 2.14 – Методи класифікації пакетного мережевого трафіку

Історично склалося так, що багато додатків використовують "добре відомі" порти на своїх локальних хостах. В цьому випадку завдання класифікатора полягає в пошуку TCP SYN-пакетів, щоб визначити серверну сторону нового клієнт-серверного TCP-з'єднання. Потім, щоб зробити висновок про додаток, який генерує трафік, розглядається номер порту призначення пакета в списку зареєстрованих портів IANA. UDP використовує порти схожим чином, але без встановлення з'єднання.

До безсумнівних переваг методу відносяться простота його реалізації і висока швидкість роботи. Однак цьому методу властивий ряд недоліків. По-перше, деякі додатки можуть не мати своїх портів, зареєстрованих в IANA, наприклад пірінгові додатки, такі як Napster і Kazaa. Додатки можуть використовувати відмінні від добре відомих портів, щоб обійти обмеження контролю доступу в операційній системі. Наприклад, непрівілейовані користувачі на UNIX-подібних системах можуть примусово запустити HTTP-сервери на портах, відмінних від 80. Також в деяких випадках порти сервера надаються динамічно в міру необхідності. В деяких випадках шифрування IP рівня може сплутати TCP та UDP заголовки, що робить неможливим визначення фактичного номера

порту. В праці [61] показано, що на основі аналізу портів не вдається визначити 30-70% потоків Інтернет трафіку.

Щоб уникнути повної залежності від номерів портів і зібрати відомості про використаний протокол, багато сучасних комерційних продуктів використовують відновлення стану сеансу і прикладну інформацію із вмісту кожного пакета. При цьому виділяють чотири різні рівні перевірки.

Перший рівень перевірки полягає в пошуку сигнатур, визначених для відомого протоколу, в корисному навантаженні прикладного рівня. Так, наприклад, HTTP-пакет починається з команди, наступної за URL і версією протоколу, у той час як більшість Edonkey пакетів має поля, що містять розмір корисного навантаження.

Другий рівень перевірки – синтаксичний. Він може розглядатися як більш точна версія сигнатурної перевірки, оскільки спрямований на перевірку правильності переданих даних з синтаксичної точки зору (наприклад, передбачається, що корисне навантаження HTTP повинна містити HTTP заголовки). У цьому випадку необхідно декодувати всі поля, що містяться у повідомленні.

Третій рівень контролю пов'язаний з протоколом відповідності. Наприклад, цей рівень контролює, що на HTTP GET запит від клієнта дійсно надходить відповідь від сервера. Така форма контролю є більш точною, оскільки вона може перевіряти відповідно до специфікації реальну поведінку протоколу.

Четвертий рівень контролю відноситься до семантики даних. Наприклад, алгоритми цього рівня здатні перевірити, чи об'єкт, що передається по протоколу HTTP, є зображенням або іншою формою змісту. Такий контроль дуже корисний для виявлення "тунелів", в яких додаток використовує інший протокол для транспортування даних. На даний час це найбільш неформалізований рівень.

Аналіз корисного навантаження дозволяє класифікувати трафік

незалежно від його порту призначення, однак цей аналіз істотно завантажує пристрій ідентифікації трафіку. Крім того, застосування цього підходу може бути ускладнене або неможливе, коли йде мова про запатентовані протоколи або зашифрований трафік.

Значного прогресу у вирішенні проблеми класифікації трафіку вдалося досягти в рамках підходу, основаного на аналізі потоку пакетів (flow-based techniques). Як потік розглядається ряд пакетів, що поділяють однаковий кортеж з п’яти елементів: IP-адреса джерела та отримувача, номер портів джерела і отримувача, номер протоколу. Аналіз унікальних статистичних характеристик двонаправленого потоку пакетів методами інтелектуального аналізу даних дозволяє отримати вирішальні правила для класифікації трафіку в режимі реального часу (рис. 2.15).

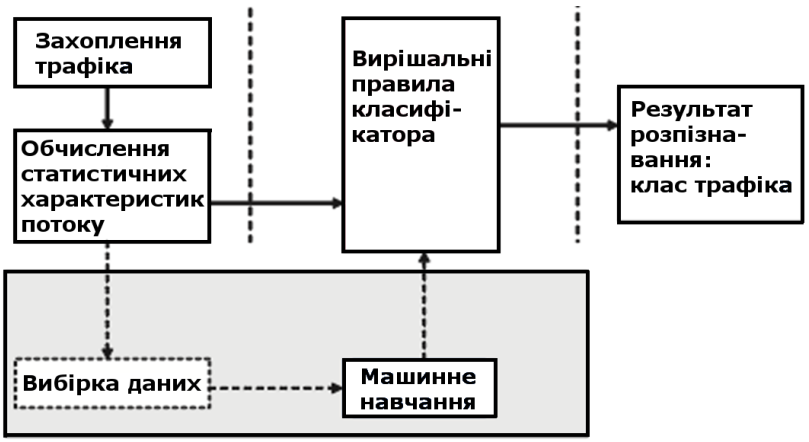


Рисунок 2.15 – Здатний навчатися класифікатор трафіку

Двонаправлений потік пакетів характеризується великою кількістю ознак розпізнавання: кількість пакетів та байтів в прямому/зворотному напрямках потоку; відношення кількості пакетів до кількості байтів корисного навантаження в прямому/зворотному напрямках; середнє значення, мінімальне значення, перша та третя квартілі, медіана та дисперсія розміру корисного навантаження (в байтах) для вхідних/вихідних пакетів двонаправленого потоку; відношення кількості пакетів малого розміру (до 50 байтів корисного навантаження) до

загальної кількості пакетів в прямому/зворотному та в обох напрямах; відношення кількості пакетів великого розміру (більше 1300 байтів корисного навантаження) до загальної кількості пакетів в прямому/зворотному та в обох напрямах; мінімальне, максимальне та середнє значення тривалості часового інтервалу між прибуттям пакетів в прямому/зворотному напрямах; відношення кількості пакетів без корисного навантаження до загальної кількості пакетів в прямому/зворотному та в обох напрямах; кількість прапорців ACK / PSH в потоці прямого/зворотного напряму та інші.

Значне розширення алфавіту класів трафіку у зв’язку з появою нових видів додатків, таких як Google Talk, Facebook, Cloud Computing, Big Data, Yahoo Messenger та інших обумовлює необхідність розширення та оптимізації словника ознак розпізнавання. При цьому процес формування репрезентативної вибірки зразків деяких типів трафіку має специфіку і різну трудомісткість, що може виражатися в неоднакових розмірах вибірки. Тому скорочення часу аналізу трафіку є актуальною проблемою, вирішення якої потребує розроблення алгоритмів машинного навчання, здатних будувати високодостовірні вирішальні правила за умов незбалансованості класів.

Навчальні набори даних можуть бути сформовані в процесі трасування трафіку утилітою TcpDump з наступним розщепленням потоків і обчисленням ознак розпізнавання за допомогою утиліти NetMate. Апріорна класифікація зразків трафіку комп’ютерних додатків може бути здійснена за результатами моніторингу сокетів утилітою CurrPorts (Windows) або NetActivityViewer (Linux).

З метою забезпечення репрезентативності навчальних вибірок розглядаються різні сценарії мережевої взаємодії. Наприклад, для трасування трафіку SSH необхідно здійснювати інформаційний обмін між різними мережами / підмережами, де задіяні комп’ютери з різними операційними системами (Windows, Linux, Mac) (рис. 2.16). При цьому в

навчальну вибірку SSH варто включити зразки трафіку передачі даних (SSH-сервіси SCP та SFTP), HTTP протокол завантаження веб-сторінок (один SSH-тунель для кожної HTTP-сесії) і сервіси віддаленого керування операційними системами та активним мережевим обладнанням.

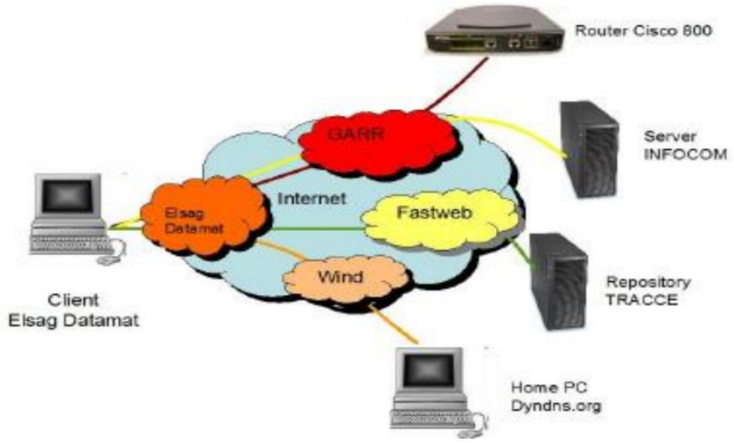


Рисунок 2.16 – Схема формування вибірок SSH трафіку

На рис. 2.17 показано конфігурацію, в якій здійснюється генерація та трасування репрезентативного трафіку SHH-сервісів.

Рисунок 2.17 – Схема формування вибірок VoIP трафіку

В запропонованій конфігурації декілька серверів, персональних комп’ютерів та маршрутизатор знаходяться в доменах різних провайдерів ISP (Internet Service Provider). При цьому аналіз трафіку здійснюється через дзеркальні порти (Mirror Port) межових комутаторів (Edge Switch).

Для аналізу VoIP трафіку репрезентативні навчальні вибірки повинні містити зразки трафіку, сформовані за участі всіх доступних кодеків (G.711, G.723.1, G.726, G.729A, GSM, iLBC і Speex), які впливають на довжину IP пакету і час прибуття. При цьому збір даних варто здійснювати за участі комерційних програмних телефонів, традиційних телефонів, під'єднаних до PSTN (Public Switched Telephone Network) та IP-телефонів, зареєстрованих в публічних SIP-акаунтах .

У випадку обробки викликів між програмним і традиційним телефоном збір RTP потоків здійснюється на стороні IP-телефону. При цьому місцем прослуховування трафіку може бути так само дзеркальний порт межового комутатора. Оскільки шифрування не забезпечує цілковитої конфіденційності внаслідок витоку інформації про активність користувача при спостереженні за ознаками потоку пакетів, то можуть знадобитися додаткові заходи, які називаються маскуванням трафіку і застосовуються на кінцях відповідного маршруту, що пролягає в незахищеній мережі (рис. 2.18).

Рисунок 2.18 – Формування захищеного каналу передачі трафіку

Методи маскування трафіку покликані модифікувати потік пакетів таким чином, щоб звести кількість інформації, яка "просочується", до мінімуму. Модифікація потоку пакетів здійснюється як за рахунок вставки

(padding) в потік фіктивних повідомлень та затримок, так і шляхом фрагментації (fragmentation) потоку, які призводять до зміни довжини, кількості пакетів та інтервалів часу між їх надходженням. На рис. 2.19 показана послідовність модифікації початкового трафіку перед його надходженням в незахищену мережу.

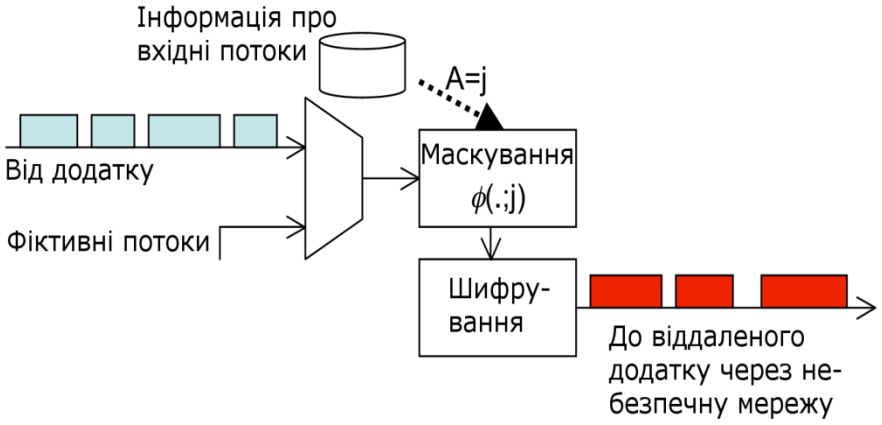


Рисунок 2.19 – Загальна схема модифікації потоку трафіку

Кожний вхідний потік $A=j$ маскується потоком $\varphi(j)$ та шифрується. Наявність апріорної інформації про статистичні властивості потоків заданого алфавіту класів у вигляді функції розподілу ймовірностей для словника інформативних ознак дозволяє оптимізувати процес маскування. При цьому ознаки результуючого потоку можуть спостерігатися в точці моніторингу. На стороні отримувача відбуваються зворотні операції (розшифрування та демаскування).

Найпростіший метод маскування полягає у встановленні незмінного розміру пакетів даних, який дорівнює максимальному розміру MTU (Maximum Transmission Unit), що підтримується протоколом. Проте для багатьох протоколів застосування такого підходу призведе до збільшення обсягу даних, що передаються по каналу, в два і більше разів. Найбільш відомі такі тривіальні підходи до модифікації довжини пакетів:

1) лінійна вставка (Linear padding), де всі довжини пакетів збільшені до найближчого кратного 128 чи MTU, дивлячись що менше;

2) експоненційна вставка (Exponential padding), де всі довжини пакетів збільшені до найближчої степені числа 2, або довжини MTU, дивлячись що менше;

3) вставка Миші-Слони (Mice-Elephants padding) – якщо довжина пакету менша або рівна 128, то довжина пакету збільшиться на 128, інакше пакет доповнюється до довжини MTU;

4) килимок для MTU (Pad to MTU), де всі довжини пакетів збільшені для досягнення довжини MTU;

5) випадкова вставка до MTU (Packet Random MTU padding) – якщо M є довжиною MTU, а L є довжиною вхідного пакету, то кожний пакет буде збільшено на значення, що обирається випадково з множини $\{0, 8, 16, \ldots, M - L\}$.

Перелічені підходи використовуються в багатьох архітектурах захищеного зв’язку, основаних на протоколах SSH, TLS і IPSec. В праці [62] на прикладі задачі ідентифікації трафіку завантаження веб-сторінок показано, що застосування цих протоколів не забезпечує надійного маскування, а точність класифікатора Байєса при ідентифікації трафіку становила 98%.

Оскільки у загальному випадку маскуванню підлягає не лише реальна довжина пакетів, але й інші характеристики потоку пакетів, то задача маскування може бути сформульована як задача пошуку шаблону трафіку певного класу і шаблону дій необхідних для його трансформації в шаблон захищеного трафіку. Шаблон трафіку можна описати системою обмежень на ймовірність появи значень ознак відповідного потоку пакетів, а шаблон дій – характеристиками фіктивного потоку, що додається до початкового, та параметрами фрагментації. На рис. 2.20 показано схему модифікації трафіку.

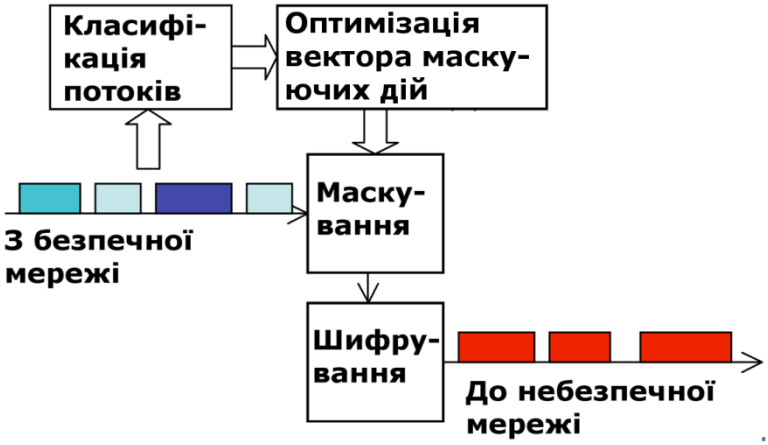


Рисунок 2.20 – Маскування основане на шаблонах

Класифікатор трафіку в схемі (рис. 2.20) дозволяє ідентифікувати шаблон трафіку, який відповідає одному з апріорно відомих класів. Відмінності даного шаблону від шаблону ідеального зашифрованого трафіку є вхідною інформацією для алгоритму оптимізації шаблону дій щодо модифікації початкового потоку трафіку. Під час оптимізації крім наближення шаблону трафіку до ідеально зашифрованого повинна вирішуватися задача мінімізації накладних витрат, пов’язаних як з процесом маскування, так і з процесом транспортування трафіку надлишкового обсягу. При цьому типовий модуль маскування включає такі процедури:

1) формування черги пакетів перед надсиланням;

2) вибір пакетів з черги згідно процедури таймінгу (для регулювання інтервалів часу між надходженням пакетів);

3) керування довжиною пакетів відповідно до методу вставки фіктивних пакетів;

4) генерація фіктивних пакетів.

5) формування результуючого захищеного трафіку.

Підвищення ефективності маскування та продуктивності основних протоколів є актуальними для реалізації анонімних комунікацій, наприклад при розробці сервісів миттєвого обміну повідомленнями та VoIP в соціальних мережах. В соціальних мережах маскування трафіку

може приховати від зловмисника інформацію про те хто з ким спілкується.

Застосування різних методів маскування трафіку дозволяє покращити підтримку конфіденційності інформації, проте такі заходи можуть ввести в оману механізми пріоритезації та фільтрації трафіку, що може призвести до зниження QoS для високопріоритетних додатків.

Останнім часом набуває поширення використання засобів децентралізованої технічної анонімізації, яка здатна приховати, як контент і тип трафіку, так і IP-адресу та реквізити користувача. Порівняно з централізованими рішеннями, такими як http-проксі-сервери, SOCKS-проксі-сервери, VPN-сервіси, SSH-тунелі, децентралізовані системи характеризуються підвищеною відмовостійкістю сервісів анонімізації навіть при великій кількості користувачів та вищим ступенем захищеності від цензури. До найбільш популярних децентралізованих сервісів анонімізації відносяться I2P (Invisible Internet Project) та Tor (The Onion Router) мережі.

I2P – це анонімна оверлейна (поверх Інтернету), самоорганізована розподілена мережа, яка використовує модифікований DHT Kademlia, але відрізняється тим, що зберігає в собі хешування адреси вузлів мережі, зашифровані AES IP-адреси, а також публічні ключі шифрування, причому з'єднання теж зашифровані. При пересиланні повідомлення використовується чотири рівні шифрування (наскрізне, часникове, тунельне, а також шифрування транспортного рівня), перед шифруванням у кожен мережевий пакет автоматично доповнюється невеликою випадковою кількістю випадкових байтів (вставка фіктивних даних). Мережа надає додаткам простий транспортний механізм для анонімного і захищеного пересилання повідомлень один одному. Основним недоліком є низька швидкість передачі даних та неможливість звернення до будь-яких ресурсів Інтернет. Працюючи з I2P можна звертатися лише до I2P-ресурсів (I2P-сайтів, пошти, трекерів тощо.).

Tor – це система проксі-серверів, в якій клієнт з'єднується з Інтернетом через ланцюжок вузлів. Як правило, ланцюжок складається з трьох вузлів, кожному з них невідомі адреси клієнта і ресурсу одночасно. Крім того, Tor шифрує повідомлення окремо для кожного вузла, а відкритий трафік видно тільки вихідному вузлу. Tor складається з десятків керуючих (вхідних) вузлів, біля десятку тисяч вузлів-посередників, та декількох тисяч вихідних вузлів. На рис. 2.21 показано спрощену схему взаємодії клієнта з сервером через Tor-мережу.

Рисунок 2.21 – Спрощена схема взаємодії клієнта з сервером через Tor-мережу

Трафік у зворотному напрямі надходить у відкритому вигляді, шифрується на вихідному вузлі на тимчасових симетричних ключах і передається далі по ланцюжку.

Застосування Tor лишає можливості зловмиснику, який відслідковує інтернет-з’єднання клієнта, визначити які сайти були відвідані, що лишає можливості дізнатися фізичне розташування клієнта. Сервіс Tor працює з додатками, які використовують TCP-протокол (веб-браузери, клієнтські програми служб миттєвого обміну повідомленнями, програми віддаленого входу в систему та ін.). Однак вихідний трафік вихідних вузлів Tor може бути перехоплений і піддаватися аналізу (рис. 2.22). Індійський дослідник, професор Самбудхо Чакравартi показав, що можна

деанімізувати до 80% трафіку, що проходить крізь Tor, використовучи технологію NetFlow, яка вбудована в маршрутизатори Cisco, та спеціальну програму аналізу Tor-трафіку. Достатньо одного потужного сервера, щоб в процесі порівняння потоків вихідного трафіку з потоками вхідного трафіку Tor відслідкувати до 39% випадково створених ланцюжків серверів Tor.

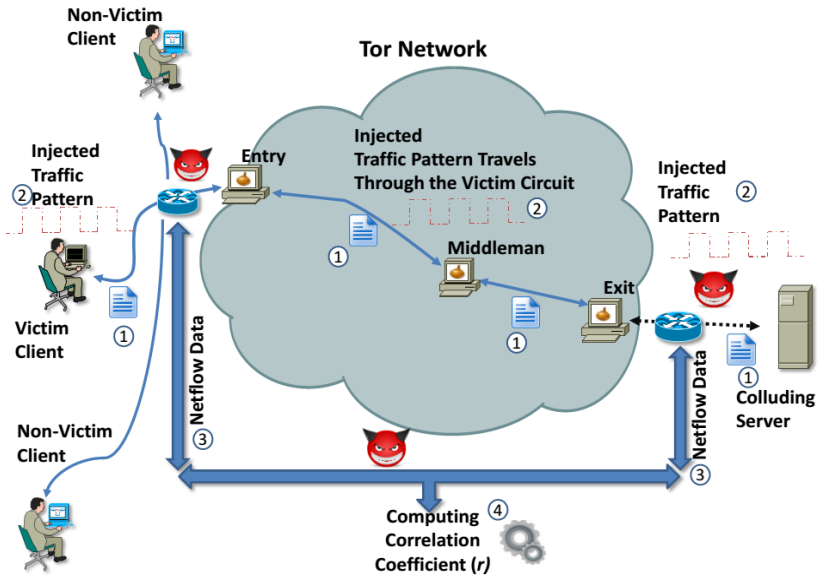


Рисунок 2.22 – Схема аналізу трафіку Tor системи

Джерелом розкриття відомостей про користувача (ідентифікація користувача в Інтернет) може бути інформація, яку передає програмне забезпечення в обхід блокуючих міжмережевих екранів, проксі-серверів тощо, для забезпечення нормальної та ефективної роботи в складних мережевих умовах. Це, як правило, різного роду дані, передача яких передбачена специфікацією до програмного продукту. Наприклад, веб-браузер містить такі функціональні компоненти і технологічні категорії:

1) cookies – текстові файли з деякими даними, що їх зберігають прикладні програми для різних задач, наприклад, аутентифікації. Розкриття анонімного клієнта настає, якщо він спочатку відвідав ресурс через відкритий сеанс, браузер зберіг cookies, а потім користувач

з’єднався через анонімний сеанс. В результаті серверу доступно співставлення cookies і, як наслідок, деанонімізація клієнта;

2) Flash, Java – плагіни, що ґрунтуються на цих технологіях, завантажуються від імені користувача як окреме програмне забезпечення та можуть працювати в обхід проксі, зберігати свої cookies й інші налаштування;

3) відбиток (fingerprint) браузера – оглядач представляє серверу десятки категорій даних, що дає змогу сформувати унікальний цифровий відбиток браузера, за яким його можна ідентифікувати серед багатьох інших навіть в анонімному сеансі (найчастіше застосовується з метою цільової реклами);

4) скріпти JavaScript – код, що виконується на стороні клієнта, здатен накопичувати для сервера ідентифікуючу інформацію, а також за умови вразливості цільового для користувача ресурсу створює умови для проведення успішних атак на інформаційний ресурс;

5) http-referrer – за допомогою цього http-заголовку цільовий для користувач веб-сайт може визначити, ким було сформовано трафік.

Методи шифрування та маскування трафіку не здатні вплинути на анонімізацію, коли службова інформації програм є деанонімізуючою, оскільки вона, як правило, використовується в кінцевій точці транзакції, де надається вже у відкритому вигляді. У випадку веб-браузера єдиним рішенням є налаштування параметрів безпеки веб-браузера, включаючи блокування кожної із наведених категорій ідентифікації даних, та відмова під час анонімного сеансу від неперевіреного програмного забезпечення.

Таким чином, поширення мережевих додатків, які динамічно змінюють порти транспортних протоколів, використовують шифрування та інкапсуляцію трафіку в тунельний протокол, призводить до низької ефективності класифікації трафіку на основі портів чи корисного навантаження. Вирішення цих проблем пов'язується з використанням методів машинного навчання та розпізнавання образів, де, як правило,

ознаками розпізнавання виступають статистичні характеристики потоку пакетів. Проте різноманітність та параметрична невизначеність інформаційних процесів, які генерують мережевий трафік, обумовлює необхідність розгляду безлічі сценаріїв мережевої взаємодії для формування репрезентативних навчальних вибірок зразків трафіку. При цьому застосування методів маскування трафіку дозволяє знизити точність його класифікації і відповідно знизити витік конфіденційної інформації про мережеву поведінку користувача. Однак замаскований трафік, як правило, значно збільшує вимоги до пропускної здатності каналу, вводить в оману механізми і політики QoS. Забезпечення анонімності поведінки та місця знаходження автора за рахунок децентралізованих систем типу Tor та I2P так само збільшує накладні витрати і сповільнює мережевий трафік користувача, однак деанонімізація вимагатиме на декілька порядків більше витрат та зусиль. Тому залишається відкритим питання компромісу між захищеністю каналу зв’язку та накладними витратами для її забезпечення

2.3 Виявлення атак та керування безпекою в інфокомунікаційній системі

Ефективне керування ресурсами інформаційно-комунікаційної системи обов’язково включає в себе забезпечення надійності функціонування та безпеки інформації. Існує необхідність виявляти та протидіяти інформаційним впливам протиправного та деструктивного характеру, які ще називають атаками або вторгненнями. Для цього використовують системи виявлення атак IDS (Intrusion Detection System), які займаються аналізом використання доручених їм ресурсів і, у випадку виявлення яких-небудь підозрілих чи просто нетипових подій, здатні вживати деякі самостійні дії щодо виявлення, ідентифікації та усунення їх причин.

В процесі розвитку комп'ютерних та мережевих технологій список можливих типів мережевих атак на інформаційно-комунікаційні системи постійно розширюється. В загальному випадку можна виділити чотири групи найбільш поширених атак:

1) U2R (user-to-root) атаки передбачають отримання зареєстрованим користувачем привілеїв локального суперкористувача (адміністратора);

2) R2L (remote-to-local) атаки характеризуються отриманням доступу незареєстрованого користувача до комп'ютера з боку віддаленої машини;

3) Probe атаки полягають в скануванні мережевих портів з метою отримання конфіденційної інформації;

4) DoS (denial of service) атаки – це мережеві атаки, спрямовані на виникнення ситуації, коли в системі, що атакується відбувається відмова в обслуговуванні.

Залежно від джерела виявлення атак розрізняють системи рівня host-based (HBIDS), мережеві IDS (NIDS – network intrusion detection), а також системи рівня додатків (APIDS) і гібридні IDS (HIDS), які поєднують комбіновані методи. Перші ідентифікують вторгнення, аналізуючи події і трафік, що надходить на окремий комп'ютер, в той час як інші – досліджують мережевий трафік. Системи рівня додатків, як правило, розташовуються між веб-сервером і, наприклад, SQL-сервером. Існує багато методів виявлення атак, проте на даний момент можна виділити такі основні підходи:

1) аналіз сигнатур, оснований на простому співпадінні послідовностей зі зразком атаки;

2) аналіз аномалій, оснований на контролі частоти подій чи виявленні статистичних аномалій;

3) комбінований підхід.

Аналіз сигнатур був першим методом, застосованим для виявлення атак. У вхідному пакеті проглядається байт за байтом і порівнюється з

сигнатурою (підписом) – характерним рядком програми, що вказує на характеристику шкідливого трафіку. Такий підпис може містити ключову фразу, команду або їх послідовність, які пов'язані з атакою. Системи аналізу сигнатур досить швидкі, оскільки не здійснюють повний і глибокий аналіз пакету та протоколу. При цьому вирішальні правила легко дописувати, редагувати і налаштовувати. Високий рівень підтримки комп’ютерним співтовариством у справі формування сигнатур для нових небезпек сприяє успіху застосування подібних систем при виявленні хакерських впливів на початкових етапах: простим атакам, як правило, передують дії, які легко розпізнати. Однак збільшення обсягу сигнатур значно знижує швидкість системи. В середньому кожен день з’являється біля 70 нових атак. Фізично неможливо оновлювати бази даних сигнатур за такі проміжки часу. Велика кількість атак при їх невеликих модифікаціях можуть бути не виявлені такою системою. Система основана на аналізі сигнатур здатна виявляти тільки вже відомі атаки, сигнатури яких присутні в базі.

В основі методу аналізу аномалій лежить принцип порівняння поточних кількісних показників системи (частота звернень до служб, навантаження на вузли мережі тощо) з еталонним станом, сформованим за результатами моніторингу нормального функціонування інформаційно-комунікаційної системи. Даний принцип дозволяє з успіхом виявити факт вторгнення і джерело загрози (внутрішнє або зовнішнє). У випадку, якщо джерело загрози є внутрішнім, система аналізу аномалій дозволяє визначити чи атака, яка відбувається від імені авторизованого користувача, надходить насправді від авторизованого користувача чи від зловмисника, що маскується під нього. При цьому система аналізу аномалій дозволяє виявляти незначні модифікації відомих атак, але при практичній реалізації такої системи існують ускладнення, пов’язані з виявленням та врахуванням раніше невідомих

типів атак і впливів. Тому при синтезі вирішальних правил системи аналізу аномалій необхідно вирішувати такі задачі:

1) побудова еталонної множини інваріантних ознак нормального (семантично коректного) розвитку обчислювальних процесів в умовах апріорної невизначеності впливів зовнішнього та внутрішнього середовищ;

2) встановлення шкал вимірювання інваріантних ознак;

3) виявлення інформативних інваріантних ознак;

4) синтез вирішальних правил для розпізнавання аномалій.

Для формування еталонного стану інформаційно-комунікаційної системи чи її підсистеми, як правило, використовують методи контрольованого та неконтрольованого машинного навчання.

Методи контрольованого машинного навчання використовують фіксований набір параметрів оцінки та апріорні відомості про значення цих параметрів. Період навчання та перенавчання характеризується невеликими часовими рамками. При цьому серед методів контрольованого навчання найчастіше розглядаються метод моделювання правил, описова статистика та нейронні мережі [63-65].

Методи неконтрольованого навчання використовують набір параметрів оцінки, склад яких може динамічно змінюватися. Навчання відбувається неперервно і еталонний стан з часом адаптується до нової конфігурації. Серед методів неконтрольованого навчання найбільш досліджено застосування описової статистики та моделювання множини станів [63].

В лабораторія MIT Lincoln Labs з метою дослідження ефективності різних методів машинного навчання в задачі синтезу вирішальних правил для виявлення типових атак було сформовано набір зразків поширених на той час атак. При цьому було запропоновано 41 ознаку для кожного TCP/IP з’єднання. Ознаки були розділені на три групи.

До ознак першої групи віднесено основні характеристики TCP / IP з’єднання, наприклад, тривалість зв’язку, тип протоколу, сервіс, число переданих байт від джерела до приймача і у зворотному напрямі, окремі прапорці. Деякі значення ознак визначалися із затримкою на протязі певного часового інтервалу. Друга група ознак містить статистичні характеристики трафіку, які обчислюються з використанням 2-х секундного часового вікна чи на протязі більшого часового проміжку. Характеристики поділяються на дві групи: атрибути, що відносяться до конкретного host – комп’ютеру чи до конкретного сервісу. Окремі атаки сканування портів виконуються довше, ніж 2 сек. Тому ряд ознак обробляється вікном у 100 з’єднань. До ознак третьої групи належать ознаки всередині одного окремого з’єднання. На відміну від більшості DoS-атак і сканування портів, R2L та R2U атаки характеризуются окремими нетривалими проявами до окремого комп’ютера. В той час як DoS – атаки та Probing ініціалізують множинні з’єднання в короткий проміжок часу.

Впровадження стеку протоколів IPv6, поява хмарних сервісів і мережевих додатків провокують появу нових схем для реалізації мережевих атак, що обумовлює необхідність розширення і перегляду словника ознак.

У загальному випадку система виявлення вторгнень має ієрархічну організацію, в якій доцільно окремо розглядати локальну і глобальну архітектури. В рамках локальної архітектури реалізуються елементарні складові, які потім можуть бути об'єднані для обслуговування корпоративних систем. Основні елементи локальної архітектури і зв'язки між ними показано на рис. 2.23. Сенсори, виконані у вигляді автономних агентів, здійснюють первинний збір даних. Реєстраційна інформація може вилучатися із системних та прикладних журналів або безпосередньо від ядра операційної системи. Також цю інформацію можна отримати з мережі за допомогою відповідних механізмів активного мережевого

обладнання або шляхом перехоплення пакетів за допомогою встановленої в режим моніторингу мережевої карти.

Рисунок 2.23 – Основні елементи локальної архітектури систем виявлення атак

На рівні агентів (сенсорів) може виконуватися фільтрація даних з метою зменшення їх обсягу. Це вимагає від агентів деякого інтелекту, проте дозволяє розвантажити інші компоненти системи. Агенти передають інформацію в центр розподілу, де відбувається подальша фільтрація, приведення до єдиного формату, збереження в базі даних та передача даних для статистичного і експертного аналізу.

Якщо в процесі статистичного або експертного аналізу виявляється підозріла активність, відповідне повідомлення направляється до блоку прийняття рішень, де визначається виправданість тривоги і обирається спосіб реагування. Глобальна архітектура відображає організацію однорангових і різнорангових зв'язків між локальними системами виявлення атак (рис. 2.24). На одному рівні ієрархії розташовуються компоненти, що аналізують підозрілу активність. Наприклад, на хості можуть розташовуватися підсистеми аналізу поведінки користувачів і

додатків. Їх може доповнювати підсистема аналізу мережевої активності. Коли один компонент виявляє щось підозріле, то в багатьох випадках доцільно повідомити про це сусідам або для вжиття заходів, або для посилення уваги до певних аспектів поведінки системи.

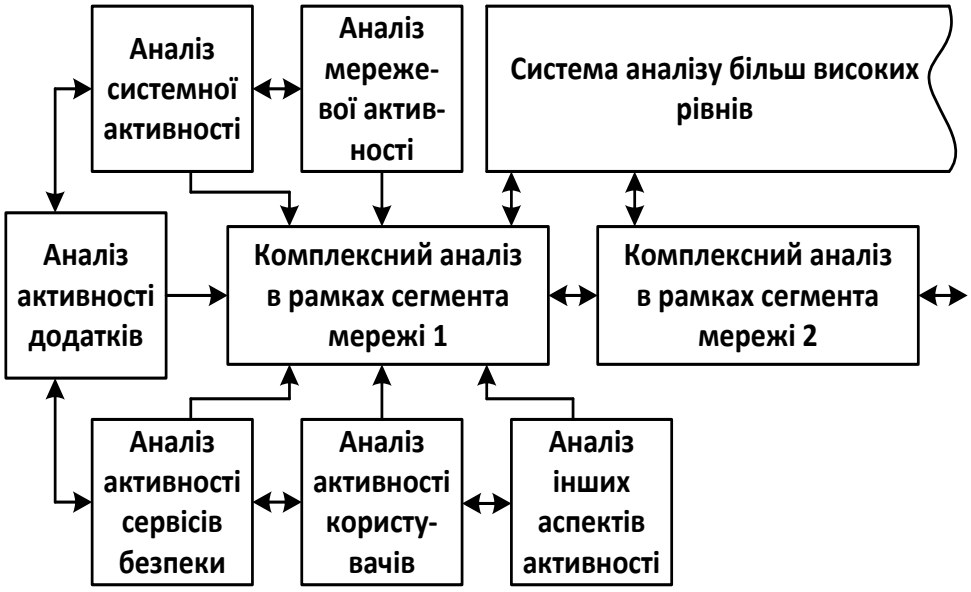


Рисунок 2.24 – Глобальна архітектура систем виявлення атак

Різнорангові зв'язки використовуються для узагальнення результатів аналізу та отримання цілісної картини того, що відбувається. Іноді у локального компонента недостатньо підстав для порушення тривоги, але в сукупності підозрілі ситуації можуть перевищити поріг підозрілості. Цілісна картина, можливо, дозволить виявити скоординовані атаки на різні ділянки інформаційної системи та оцінити збиток в масштабі організації.

Група IDWG напрацювала специфікацію формату IDMEF (Intrusion Detection Message Exchange Format) – формат обміну даними між компонентами IDS. Формат IDMEF використовується для передачі попереджувальних повідомлень про підозрілі події між системами виявлення атак. Даний формат повинен забезпечити сумісність між

комерційними і вільно поширюваними IDS і можливість їх спільного використання для забезпечення найвищого рівня захищеності.

Забезпечення захищеності інфокомунікаційної системи потребує введення надлишковості конфігурації системи, що пов’язано з використанням додаткових засобів захисту інформації, які споживають певний обсяг системних ресурсів. Тому існує потреба в розробці адаптивної системи керування захистом інфокомунікаційної системи, яка буде здатна змінювати рівень захищеності об’єктів системи залежно від їх інформаційної цінності. При цьому показник інформаційної цінності ресурсів може визначатися реальною вартістю чи обсягом збитків у випадку знищення чи втрати конфіденційності інформації, що в них міститься. Однак цінність інформації змінюється з плином часу. Залежно від динаміки і типу інформації виділяють такі випадки:

– цінність інформації стаціонарна в часі (бази даних, інформація в яких актуальна на протязі тривалих періодів часу);

– цінність інформації постійно збільшується (бази даних в момент накопичення інформації);

– цінність інформації постійно зменшується (бази даних, актуальність інформації в яких падає);

– цінність інформації має верхній екстремум (інформація, яка в певний момент часу змінює свій статус, наприклад – про розробки, які патентуються, про передвиборчу кампанію);

– цінність інформації має нижній екстремум (теоретично можливий випадок).

Адаптивна система керування захистом повинна динамічно оцінювати цінність інформаційних ресурсів і адаптивно змінювати захищеність інформаційно-комунікаційної системи. Архітектура такої системи може мати вигляд, показний на рис. 2.25 [66]. До засобів збору інформації та аналізу стану системи відносяться класичні системи виявлення вторгнень і засоби захисту (антивіруси, міжмережеві екрани,

апаратні засоби захисту та контролю доступу) та засоби аналізу захищеності інфокомунікаційної системи. Ці засоби, при виявленні підозрілої активності, яку не можна достовірно ідентифікувати, відносять її до потенційних атак.

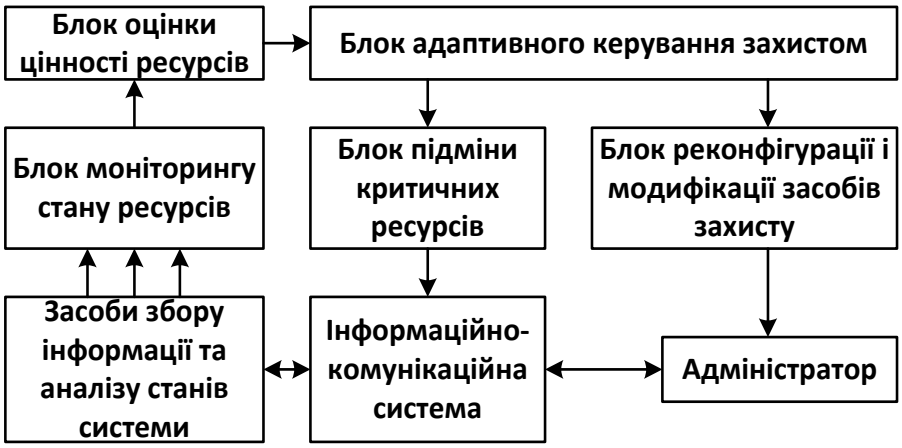


Рисунок 2.25 – Архітектура системи адаптивного керування захищеністю інфокомунікаційної системи

Блок моніторингу станів ресурсів інфокомунікаційної системи отримує інформацію від засобів збору інформації та аналізу станів системи. При цьому кожен запис про підозрілу активність конвертується в певний формат, який включає дані про суб'єкта (ініціатора події), параметри дій суб'єкта по реалізації атаки, можливі цілі, час, місце, задіяні засоби і ступінь успішності атаки. Крім цього завданням блоку моніторингу є формування ймовірностей вторгнень.

Блок оцінки цінності ресурсів виконує подальший аналіз інформації, отриманої від блоку моніторингу безпеки. Для кожного ресурсу, обчислюється непрямий показник його інформаційної цінності. Ці дані є базовими для адаптивної зміни рівня захищеності ресурсів з урахуванням цінності інформації, яку вони обробляють чи зберігають.

Блок адаптивного керування захищеністю ресурсів інформаційно-комунікаційної системи диференційовано для кожного ресурсу виконує порівняння поточної цінності ресурсу зі значеннями, отриманими на попередньому кроці аналізу, і визначає необхідний рівень захищеності для кожного з них. Потім інформація передається блоку реконфігурації і модифікації засобів захисту для налаштування нових параметрів засобів захисту або зміни їх конфігурації. Крім того, даний блок забезпечує необхідну інформацію для блоку підміни критичних ресурсів, що дозволяє додатково підвищити захищеність критичних даних.

Блок реконфігурації і модифікації засобів захисту здійснює цілеспрямовану зміну структури системи захисту інформації шляхом перерозподілу засобів захисту від об'єктів, що мають мінімальну інформаційну цінність, або об'єктів, що не піддаються в даний момент атакам, до об’єктів, на які спрямовано в даний момент дистуктивний вплив.

Блок підміни критичних ресурсів виробляє емуляцію вразливостей, системних відомостей і критичних інформаційних ресурсів, на основі даних блоку визначення потенційної цінності ресурсів. При цьому залежно від характеру дій і цілей зловмисника можуть застосовуватися як приховування та дезінформація, так і повна підміна критичних даних.

Адміністратор виконує загальний контроль всієї системи адаптивного керування, відповідає за зв'язок і взаємодію інформаційно-комунікаційної системи із засобами збору інформації та оцінки стану системи. Крім того, він приймає остаточне рішення про емуляцію вразливостей, реконфігурацію системи захисту і подальші дії щодо порушників.

Таким чином, в сучасних інфокомунікаційних системах все частіше впроваджуються засоби інформаційної безпеки, основані на ідеях і методах машинного навчання та розпізнавання образів. Архітектура системи виявлення вторгнень набуває ієрархічної структури, на всіх

рівнях якої знаходяться здатні до взаємодії інтелектуальні автономні агенти. При цьому керування безпекою інфокомунікаційної системи полягає у своєчасній реакції на мережеві атаки та забезпеченні економічно доцільного рівня захищеності об’єктів системи відповідно до їх інформаційної цінності.

3 ІНФОРМАЦІЙНИЙ СИНТЕЗ СИСТЕМИ КЕРУВАННЯ ІНФОКОМУНІКАЦІЙНИМ СЕРВІСОМ

3.1 Узагальнений критерій ефективності здатної навчатися системи керування інформаційно-телекомунікаційним середовищем

Сучасні підходи до керування інформаційно-телекомунікаційними системами передбачають забезпечення якості послуг, що сприймаються споживачами, виконання вимог договорів та укладання угод про рівень обслуговування. Нові методи управління в сфері телекомунікацій сьогодні більше концентруються на забезпеченні зовнішніх умов функціонування (інформаційна безпека, нормативні акти), а також на управлінні діяльністю сервісного оператора (провайдера).

З метою формалізації задачі синтезу системи керування інформаційно-телекомунікаційними сервісами та відповідною IT-інфраструктурою обирається критерій, здійснюється постановка та рішення оптимізаційної задачі. Наприклад, задача мінімізації енерговитрат при функціонуванні інформаційно-телекомунікаційної системи за умов забезпечення заданої якості обслуговування користувачів.

Інформаційно-телекомунікаційні системи можуть характеризуватися такими показниками, як доступність, надійність, продуктивність, конфіденційність, масштабованість, вартість сервісів, прибутковість, час обслуговування, функціональність, якість обслуговування та інші. Ці критерії можна розглядати як однопараметричні (часткові) критерій ефективності, які не дають достатньо повного уявлення про ефективність системи у цілому. Тому спроба вибору кращого варіанту проектування системи одночасно за декількома частковими критеріями зазвичай позбавлена змісту, оскільки,

як правило, поліпшення одного критерію супроводжується погіршенням, принаймні, одного іншого критерію. Все це вказує на необхідність використання узагальнених критеріїв, що пов'язують у необхідних пропорціях основні, найважливіші часткові критерії системи.

При проектуванні систем керування інформаційно-телекомунікаційними сервісами та відповідною IT-інфраструктурою важливо не тільки забезпечити необхідні технічні характеристики, але і врахувати затрати на їх одержання. Цій вимозі відповідають техніко-економічні критерії, найбільш поширеним серед яких є критерій вигляду

$$Q = (\text{ефект}) / (\text{витрати}). \qquad (3.1.1)$$

Цей критерій ефективності можна розглядати як основу синтезу нових критеріїв ефективності. При цьому якщо оцінювати технічну ефективність лише з точки зору отримуваного прибутку, то загальну ефективність системи керування можна подати у вигляді:

$$Qc = (Cv - Cs) / Cvi, \qquad (3.1.2)$$

де *Cv* – результат використання системи (реальний прибуток); *Cs* – витрати на створення та експлуатацію системи; *Cvi* – результат застосування системи при виконанні всіх функцій і відсутності витрат на їх здійснення (ідеальний прибуток).

До задач, що виконує система керування інформаційно-телекомунікаційною інфраструктурою, відносяться виконання контролю працездатності розподіленої мережевої інфраструктури, спостереження за станом серверів і систем зберігання, автоматизації рутинних операцій керування центрами обробки даних, забезпечення доступності та належної продуктивності бізнес-додатків, які звертаються до хмарних ресурсів та використовують мобільні інтерфейси та інші. При цьому

ефективність керування визначається багатьма факторами, найбільш негативними з яких є апріорна невизначеність поведінки вузлів та додатків на гетерогенних ресурсах, непередбачувана динаміка попиту/ пропозиції / доступності ресурсів. Однак у великих інформаційно-телекомунікаційних системах засоби моніторингу та автоматизації керування генерують та накопичують великі обсяги даних, що містять цінну інформацію про можливості підвищення ефективності функціонування.

Практично неперервно і в дуже великих кількостях генеруються дані лог-файлів, різноманітні метрики стану компонентів інфраструктури, інформація про події та інша телеметрія від засобів моніторингу та керування серверами, системами зберігання, мережевої та мобільної інфраструктурами і додатками. Виникає потреба у розвинутому аналітичному інструментарії, що дозволить агрегувати дані систем керування інфраструктурою в різних доменах, виявляти залежності і кореляції, передбачати збої і падіння продуктивності, надавати інформацію для оперативного і найбільш ефективного вирішення проблем, прогнозування потреб сервісів в різних ресурсах. Тому система керування повинна містити у своєму складі систему підтримки оптимальних рішень по відновленню, забезпеченню продуктивності і розвитку інформаційно-телекомунікаційної інфраструктури. При цьому інформаційна спроможність системи підтримки прийняття рішень є основною складовою загальної ефективності системи керування.

Згідно праці [67] І. В. Кузьміна узагальнену ефективність системи керування можна визначити її двома складовими: інформаційна спроможність системи та зведена вартість створення, експлуатації, зберігання та ліквідації системи. При цьому узагальнений функціонально-статистичний критерій ефективності має такий вигляд:

$$E_{I,C} = K_I / K_{I0}, \qquad (3.1.3)$$

де K_I – узагальнена функціонально-статистична характеристика реальної системи:

$$K_I = I_{\max}/C, \tag{3.1.4}$$

де $I_{\max}$ – максимальна інформаційна спроможність системи; С – зведені витрати на створення, експлуатацію та утилізацію системи.

Узагальнена функціонально-статистична характе-ристика потенційної (ідеальна) системи визначається як

$$K_{I0} = I^0_{\max} / C_{\min}, \tag{3.1.5}$$

де $I^0_{\max}$ – максимальна інформаційна спроможність потенційної системи; $C_{\min}$ – зведені витрати для потенційної системи.

Максимізація інформаційної спроможності системи керування передбачає зняття невизначеності щодо задоволеності користувачів рівнем сервісу та функціонального стану компонентів або сервісів інформаційно-телекомунікаційного середовища. Вирішальні правила системи підтримки рішень можна отримати в процесі машинного навчання за архівними даними моніторингу та суб'єк-тивно-статистичних досліджень оцінки якості сервісів. При цьому процес навчання системи полягає у пошуці оптимальних значень координат вектора просторово-часових параметрів функціонування, що забезпечують максимальне значення узагальненого критерію ефективності, який з урахуванням (3.1.3) можна подати у вигляді

$$J = \frac{\overline{E}}{E_{\max}} \cdot \frac{C_{\min}}{C_{training} + C_{error}}, \tag{3.1.6}$$

де $\overline{E}$ – усереднене за алфавітом класів розпізнавання $\{X_m^o | m = \overline{1, M}\}$ значення інформаційного критерію функціональної ефективності машинного навчання; $E_{\max}$ – максимальне граничне значення інформаційного критерію; $C_{\min}$ – мінімальне граничне значення витрат оператора/провайдера, пов'язаних з експлуатацією системи керування; $C_{training}$ – значення затрат на експлуатацію системи, в тому числі витрати на формування вхідного математичного опису та вартість системних ресурсів, задіяних при навчанні (перенавчанні); C_{error} – розраховані за матрицею штрафів втрати оператора/провайдера інформаційно-телекомунікаційної системи, пов'язані з неоптимальним керуванням IT-інфраструктурою внаслідок помилкового прийняття рішень.

Серед інформаційних мір для оцінки інформаційної спроможності системи керування, що навчається, перевагу слід віддавати статистичним логарифмічним критеріям, які дозволяють працювати з навчальними вибірками відносно малих обсягів. Серед таких критеріїв найбільшого використання знайшли ентропійна міра Шеннона та інформаційна міра Кульбака.

Ентропійний критерій навчання системи керування розпізнавати реалізації класу X_m^o має вигляд [67]

$$E_m^{(k)} = \frac{H_m^{(k)} - H_m^{(k)}(\gamma)}{H_m^{(k)}}, \tag{3.1.7}$$

де $H_m^{(k)}$ – апріорна (безумовна) ентропія, що існує на k-му кроці навчання системи розпізнавати реалізації класу X_m^o; $H_m^{(k)}(\gamma)$ – апостеріорна (умовна) ентропія, що характеризує залишкову невизначеність після k-го кроку навчання системи розпізнавати реалізації класу X_m^o.

У праці [1] розглядається модифікація інформаційного критерію за Кульбаком навчання системи керування розпізнавати реалізації класу X_m^o як логарифмічне відношення повної ймовірності $P_{t,m}^{(k)}$ правильного прийняття рішень до повної ймовірності помилкового прийняття рішень $P_{f,m}^{(k)}$:

$$J_m^{(k)} = \log_2 \frac{P_{t,m}^{(k)}}{P_{f,m}^{(k)}} * [P_{t,m}^{(k)} - P_{f,m}^{(k)}]. \qquad (3.1.8)$$

Будь-який з критеріїв (3.1.7) або (3.1.8) можна виразити як функціонал від точнісних характеристик двохальтернативних рішень – помилки першого $\alpha_m^{(k)}$ та другого $\beta_m^{(k)}$ роду. При цьому допустима область визначення інформаційного критерію обмежується нерівностями $\alpha_m^{(k)} < 0{,}5$, $\beta_m^{(k)} < 0{,}5$. Для загального випадку багатоальтернативних рішень (багатокласова класифікація) критерії можна виразити як функціонали від точнісних характеристик багатоальтернативних рішень або використати методи зведення багатокласової класифікації до послідовності двохкласової.

Склад алфавіту класів розпізнавання $\{X_m^o | m = \overline{1,M}\}$ залежить від кола вирішуваних задач. В задачах, які відносяться до аналітики доступності (availability analytics), класами розпізнавання можуть бути шаблони використання або поведінки компонентів інфраструктури або сервісів. В задачах проактивного аналізу (performance analytics) інформаційно-телекомунікаційного середовища класами розпізнавання можуть бути функціональні стани та ситуації, що передують відмовам та збоям. В задачах аналітики вивчення поведінки (behavior learning analytics) система навчається розпізнавати класи нормальної та аномальної поведінки сервісу, що дозволяє згенерувати сигнал при виникненні можливих відхилень від норми. Розпізнавання відхилення від

нормального функціонування є особливо цінним інструментом в процесах керування змінами, надаючи можливість на основі метрик продуктивності, аналізу конфігурацій та моделі сервісу швидко оцінити наслідки внесення змін в компоненти інфраструктури для сервісу в цілому. В задачах аналітики потужностей (capacity analytics) як класи розпізнавання можуть розглядатися тренди потреб в ресурсах. В задачах прогнозування сприйняття якості (Quality of Experience analytics) інформаційно-телекомунікаційних сервісів класи розпізнавання характеризують рівень якості, що сприймається користувачами.

У робочому режимі навчена система керування повинна спрогнозувати функціональний стан сервісів і компонентів інформаційно-телекомунікаційного середовища, що прямо впливають на дотримання SLA-угод, при різних варіантах використання (розподілу) його ресурсів. При цьому задача планування використання ресурсів є багато-критеріальною, оскільки необхідно одночасно забезпечити мінімум енергоспоживання, обсягу невикористаних ресурсів та порушень SLA-угод. Однак ці частинні критерії є попарно суперечливими, мають різну розмірність і є нелінійними функціями контрольованих характеристик і конфігурацій інформаційно-телекомунікаційної системи. Тому слід розглянути можливість приведення (згортки) вектору перелічених частинних критеріїв до одного комплексного критерію. Найбільш загальний підхід до згортки векторних критеріїв оснований на використанні принципу зваженої суми частинних критеріїв. Розглянемо основні кроки реалізації згортки векторного критерію.

Крок 1. Нормалізація та масштабування вектора частинних критеріїв $\{k_i \mid i = \overline{1, I}\}$ за правилом

$$k_i = \begin{cases} 0,\ k_i \le k_i^{\min}; \\ \dfrac{(k_i - k_i^{\min})}{(k_i^{\max} - k_i)},\ k_i^{\min} < k_i < k_i^{\max}; \\ 1,\ k_i > k_i^{\max}; \end{cases}$$

де $k_i^{\min}$, $k_i^{\max}$ – нижня та верхня межі допустимої області значень i-го частинного критерію відповідно.

Крок 2. Визначення за допомогою певних формалізованих процедур чи із застосуванням експертних оцінок пріоритетів частинних критеріїв у вигляді вектора коефіцієнтів важливості $\{\lambda_i \mid i = \overline{1,I}\}$ кожного частинного критерію, для якого виконується умова

$$\sum_{i=1}^{I} \lambda_i = 1.$$

Крок 3. Обчислення зваженої суми частинних критеріїв за формулою

$$K = \sum_{i=1}^{I} \lambda_i k_i \qquad (3.1.9)$$

Для обчислення частинних критеріїв, пов’язаних з недотриманням SLA-угод використовуються вирішальні правила, отримані в процесі навчання системи керування.

За умови високої достовірності прогностичних вирішальних правил результат мінімізації зваженої суми частинних критеріїв (3.1.9) призводитиме до задоволення вимог клієнтів та власників інформаційно-телекомунікаційної системи.

Таким чином, використання аналітичного інструменту в складі системи керування інформаційно-телекомунікаційним середовищем

надає можливості оптимізації використання ресурсів інфраструктури в рамках реалізації окремих сервісів, планування потреб в ресурсах, уникнення відмов та збоїв, підтримки заданого рівня задоволеності користувачів, оцінки ймовірності успіху внесення різноманітних змін в середовище. Ефективність аналітичного інструменту визначає ефективність системи керування в цілому. Тому основною складовою узагальненого критерію ефективності системи керування є інформаційний критерій якості її машинного навчання. При цьому вартісна складова узагальненого критерію є допоміжною. Вона служить для контролю за доцільністю матеріальних витрат на подальше навчання (перенавчання) системи.

3.2 Алгоритми машинного навчання системи керування інформаційно-телекомунікаційним середовищем

Аналіз методів машинного навчання будемо здійснювати за умови прийняття гіпотези компактності, яка припускає, що класи утворюють компактно локалізовані підмножини в просторі об’єктів. У загальному випадку така гіпотеза є нечіткою, оскільки всі класи розпізнавання апріорно перетинаються в просторі ознак При цьому для формалізації поняття «схожості» вводиться функція відстані або метрика $d(x, y)$ в N-вимірному просторі об’єктів.

Алгоритм найближчого сусіда (nearest neighbor, NN) є найбільш простим метричним алгоритмом, який класифікує об’єкт як реалізацію того класу, якому належить найближчий об’єкт навчальної вибірки. Навчання такого класифікатора зводиться до елементарного запам’ятовування навчальної вибірки $\{y_{m,i}^{(j)} \mid m=\overline{1,M};\ j=\overline{1,n};i=\overline{1,N}\}$, де M – потужність алфавіту класів розпізнавання, n – обсяг спостережень класу X_m^o, N – кількість ознак розпізнавання. Основною перевагою

цього алгоритму є простота його реалізації, але він має ряд недоліків. По-перше, наявності викидів у навчальній вибірці призводить до нестійкості і похибок. По-друге відсутні параметри, які можна б було настроювати за навчальною вибіркою. Алгоритм повністю залежить успішності вибору дистанційної міри $d(x, y)$.

Алгоритм *k*-найближчих сусідів з метою згладжування шумового впливу викидів, буде класифікувати об'єкти шляхом голосування за *k* найближчими сусідами. Кожен із сусідів $\{y^{(j)} \mid j = \overline{1,k}\}$ голосує за віднесення реалізації образу до свого класу. Алгоритм відносить реалізацію, що розпізнається, до того класу, який набере більше число голосів. Оптимальне значення параметра *k* визначається за критерієм ковзного контролю з виключенням об'єктів по одному. Для кожного об'єкта перевіряється, чи правильно він класифікується за своїми *k* найближчими сусідами.

Міри близькості обирають виходячи з властивостей об'єктів. Квадрат евклідової відстані застосовується для надання більшої ваги більш віддаленим один від одного об'єктам. Ця відстань обчислюється таким чином:

$$d(x, y) = \sum_{i=1}^{N} (x_i - y_i)^2 .$$

Манхеттенська відстань є середньою різниці за координатами. В більшості випадків ця міра відстані призводить до таких результатів, як і Евклідова відстань. Однак для цієї міри вплив окремих великих різниць (викидів) зменшується –

$$d(x, y) = \sum_{i=1}^{N} |x_i - y_i| .$$

Степенева відстань застосовується у випадках, коли необхідно збільшити чи зменшити вагу, яка відноситься до розмірності, для якої об’єкти суттєво відрізняються. Степенева відстань розраховується за формулою:

$$d(x,y) = \sqrt[r]{\sum_{i=1}^{N}(x_i - y_i)^p}$$

де r і p – параметри, визначені користувачем.

Параметр r відповідальний за поступове зважування різниць по окремим координатам, а параметр p відповідальний за прогресивне зважування великих відстаней між об’єктами. Якщо r і p рівні двом, то ця відстань співпадає з відстанню Евкліда.

Відстань Журавльова обчислюється за формулою:

$$d(x,y) = \sum_{i=1}^{N} d_i ,$$

де

$$d_i = \begin{cases} 1, якщо |x_i - y_i| < \varepsilon \\ 0, інакше \end{cases}.$$

де ε – додатне як завгодно мале дійсне число (критерій мінімальної відстані між точками).

Міра близькості підбирається індивідуально для конкретних типів даних. Інколи адекватної міри близькості підібрати не вдається, і доводиться обирати її евристично.

У практичних задачах контролю інформаційно-телекомунікаційних систем вектор ознак функціонального стану складаються як з неперервних (кількісних), так і дискретних (категоріальних) ознак, що приймають свої значення з кінцевої неупорядкованої множини. З даними в межах номінальної шкали, в якій вимірюються категоріальні ознаки, не

можуть бути здійснені будь-які арифметичні операції, оскільки усі види числової обробки стосуються упорядкування об'єктів у кожному класі. Приведення категоріальних первинних ознак до кількісних вторинних шляхом простої нумерації значень первинних ознак рідко призводить до задовільних результатів, оскільки алгоритми будуть враховувати впорядкованість, яка не має змісту. Тому обробка даних змішаного типу в даний час викликає значні труднощі і є областю досліджень.

В праці [68] було запропоновано використання Dummy-кодування категоріальних ознак, при якому кожна первинна ознака перекодується в декілька вторинних бінарних ознак з рівно однією одиницею. Подібне перекодування дозволяє застосовувати багато класичних алгоритмів навчання, однак це сильно збільшує розмірність простору ознак, а також накладає обмеження на структуру ознакового опису спостережень. Інший підхід пов’язаний із застосуванням в алгоритмі машинного навчання метрики перекриття (відстань за Хеммінгом), при використанні якої ступінь відмінності реалізацій образів визначається кількістю неспівпадаючих значень первинних ознак. При цьому обробка змішаних даних може здійснюватися шляхом використання гетерогенної дистанційної метрики (Heterogeneous Euclidean-Overlap Metric). Гетерогенна дистанційна метрика між двома векторами-реалізаціями $\{x_i \mid i=1,N\}$ та $\{y_i \mid i=1,N\}$ обчислюється за наступною формулою

$$d(x,y)=\sqrt{\sum_{i=1}^{N} d_i^{\,2}}\,. \qquad (3.2.1)$$

Функція d_i у формулі (3.2.1) визначає відстань між двома значеннями i-ї ознаки у векторах x та y і має такий вигляд

$$d_i = \begin{cases} 1\ , \textit{якщо } x \textit{ або } y \textit{ невідомі}; \\ overlap_i(x_i, y_i)\ , \textit{якщо } i-\textit{та ознака} \\ \qquad\qquad \textit{є категоріальною}; \\ diff_i(x_i, y_i)\ , \textit{інакше}. \end{cases}$$

При наявності пропущених значень ознак, відстань між двома ознаками визначається як максимальна, тобто рівна одиниці. Врахування можливості пропущених значень в даних є досить важливим на практиці, оскільки неповнота та зашумленість вхідних даних часто обмежує застосування алгоритмів аналізу даних.

Функція перекриття для окремої категоріальної ознаки має наступний вигляд

$$overlap_i(x_i, y_i) = \begin{cases} 0\ , \textit{якщо } x_i = y_i; \\ 1\ , \textit{інакше}. \end{cases}$$

Нормалізована відстань між кількісними ознаками задана формулою

$$diff_i(x_i, y_i) = \frac{|x_i - y_i|}{\max_i - \min_i},$$

де $\max_i$, $\min_i$ – мінімальне та максимальне можливі значення, що приймає i-та ознака, відповідно.

Недоліком алгоритмів, що використовують подібну дистанційну метрику є ігнорування додаткової інформації, що описується статистикою якісних ознак. Наприклад, для категоріальної ознаки можна обчислити її частоту (число спостережень) та моду (значення, що має найбільшу частоту).

Метричні алгоритми здійснюють локальну апроксимацію вибірки, при якій обчислення відкладаються до моменту, доки не стане відомий вхідний об’єкт. Тому метричні алгоритми відносяться до методів лінивого навчання (lazy learning). Основи методів наполегливого

навчання (eager learning), що здійснюють глобальну апроксимація вибірки, було закладено в теорії багатовимірного статистичного аналізу та теорії прийняття рішень.

Суть статистичних методів навчання полягає у відновлені роздільної функції шляхом мінімізації середнього ризику помилкового прийняття рішення. Статистичні методи дозволяють побудувати вирішальні правила у випадках перетину класів розпізнавання, що має місце в практичних задачах контролю та керування слабо формалізованими процесами.

Одним із класичних статистичних методів класифікації є метод Байєса [69], відповідно до якого прийняття класифікаційних рішень здійснюється шляхом знаходження максимальної апостеріорної умовної ймовірності $p(X_m^o / x)$, що обчислена для заданого алфавіту класів розпізнавання $\{X_m^o \mid m = \overline{1, M}\}$ за формулою

$$p(X_m^o / x) = \frac{P(X_m^o) p(x / X_m^o)}{\sum_{k=1}^{M} P(X_k^o) p(x / X_k^o)},$$

де $P(X_m^o)$ – безумовна ймовірність появи класу X_m^o; $p(x / X_m^o)$ – значення функції правдоподібності (щільності розподілу ймовірностей) класу X_m^o для вхідної реалізації x.

Безумовна ймовірність появи класу X_m^o визначається як відношення числа реалізацій, що належать класу X_m^o, до загальної кількості реалізацій

$$P(X_m^o) = \frac{count(x^{(j)} \in X_m^o)}{n},$$

де $count(x^{(j)} \in X_m^o)$ – число реалізацій навчальної вибірки, що належать класу X_m^o; n – загальна кількість реалізацій образів у

навчальній вибірці.

Значення функції правдоподібності класу X_m^o для реалізації x при статистичній незалежності ознак роз-пізнавання обчислюється за формулою

$$p(x/X_m^o)=\prod_{i=1}^{N} p(x_i/X_m^o),$$

де $p(x_i/X_m^o)$ – функція щільності розподілу ймовірностей i-ї ознаки в класі X_m^o для вхідної реалізації x.

Щільності розподілу ймовірностей $p(x_i/X_m^o)$ можуть бути оцінені в рамках припущення про тип розподілу. Наприклад, може використовуватися гіпотеза про нормальний закон розподілу

$$p(x_i/X_m^o)=\frac{1}{\sigma_{m,i}\sqrt{2\pi}}\exp\left(-\frac{\left(x_i-\bar{x}_{m,i}\right)^2}{2\sigma_{m,i}^2}\right),$$

де $\sigma_{m,i}^2$ – дисперсія i-ї ознаки в класі X_m^o; $\bar{x}_{m,i}$ – математичне очікування i-ї ознаки в класі X_m^o.

Методи розпізнавання, основані на використанні формули Байєса і гіпотези про незалежність ознак, як правило, називають наївними байєсівськими класифікаторами. Перевагою байєсовських класифікаторів є простота реалізації алгоритмів класифікації, а недоліком – низька достовірність класифікації реалізацій образів у випадку обмеженого обсягу вибірок та перетину в просторі ознак класів розпізнавання. При цьому при побудові байєсівського класифікатора під етапом навчання розуміється набір статистичних даних.

Основними недоліками статистичних методів, які обмежують їх використання на практиці, є необхідність великих обсягів статистики для

апроксимації функції щільності розподілу імовірностей, виконання жорстких умов для забезпечення статистичних стійкості та однорідності та висока чутливість до репрезентативності навчальних вибірок.

У рамках найбільш універсального підходу до аналізу і синтезу здатних навчатися систем прийняття рішень, яким є геометричний підхід, набув популярності метод опорних векторів SVM (support vector machine) [70]. В основі SVM лежить ідея розділення згущення векторів гіперплощинами, що знаходяться на максимальній відстані від згущень при мінімізації зміщення реалізацій класу від опорного вектора (рис. 3.1).

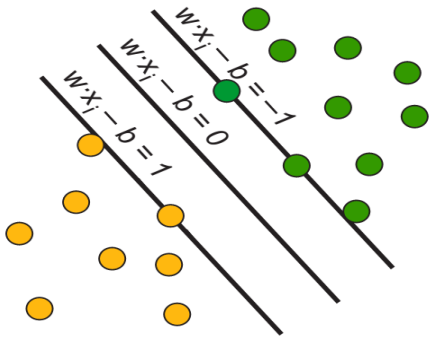


Рисунок 3.1 – Лінійне розділення образів в SVM

Задачу синтезу вирішальних правил SVM зручно формулювати для випадку двохкласової класифікації. При цьому багатокласова класифікація, як правило, зводиться до серії двохкласових класифікацій. Прямі багатокласові узагальнення SVM існують, проте навчити один складний багатокласовий класифікатор набагато довше, ніж багато простих однокласових.

Проблема лінійної нероздільності в методі SVM вирішується шляхом переходу від початкового простору ознак до нового розширеного простору за допомогою деякого перетворення. На рис.3.2 показано приклад переходу від двовимірного простору до розширеного тривимірного шляхом переносу точок на поверхню сфери.

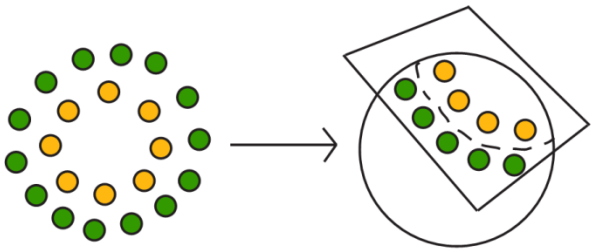


Рисунок 3.2 – Трансформація простору ознак в SVM

Аналіз рис.3.1 показує, що в новому просторі більшої розмірності вдалося розділити точки вибірок двох класів площиною.

Вирішальне правило двохкласового SVM-класифікатора має вигляд

$$a(x) = sign\left[\sum_{j}^{n} \lambda_j c_j K(x, x^{(j)}) - \rho\right], \qquad (3.2.2)$$

де ρ – параметр порогу; λ – вектор коефіцієнтів, ненульові значення компонентів якого відповідають опорним векторам навчальної вибірки; c_j – мітка класу j-го вектора навчальної вибірки, яка приймає значення із множини $\{-1,+1\}$; $K(x, x^{(j)})$ – функція ядра, що відповідає скалярному добутку векторів в деякому розширеному просторі ознак

$$K(x, x^{(j)}) = \phi(x)^T \phi(x^{(j)}),$$

де $\phi(x^{(j)})$ – функція відображення вектора $x^{(j)}$ в розширений простір, де забезпечується лінійна розділимість класів.

Навчання SVM зводиться до пошуку коефіцієнтів λ_j (множники Лагранжа) та порогу ρ, знайти які можна вирішивши таку задачу квадратичної оптимізації

$$L(\lambda) = \sum_{j=1}^{n} \lambda_j - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j c_i c_j K(x_i, x_j) \to \max_{\lambda} \qquad (3.2.3)$$

за умови

$$\begin{cases} \sum_{j=1}^{n} \lambda_j c_j = 0; \\ 0 \le \lambda_j \le C, \end{cases}$$

де C – константа регуляризації, що, як правило, обирається рівною

$$C = \frac{1}{nv}$$

де v – максимальна частка векторів навчальної послідовності, які можуть бути викидами.

Задача оптимизації (3.2.3) вирішується евристичними алгоритмами шляхом послідовного зменшення цільової функції.

Для вибору функції ядра використовують теорему Мерсера: функція $K(x,x')$ є ядром тільки тоді і тільки тоді, коли вона симетрична, $K(x,x') = K(x',x)$, і невід'ємно визначена. Однак вибір ядра для конкретних вхідних даних досі залишається нетривіальною задачею. Повний перебір всіх можливих ядер ускладнений повільністю навчання SVM.

Основним недоліком цього методу, що обмежує його застосування для задач класифікаційного аналізу функціонального стану інформаційно-телекомунікаційної системи є модельність алгоритму, обумовлена ігноруванням апріорного перетину класів розпізнавання в просторі ознак.

На усунення недоліків відомих методів машинного навчання спрямовано інформаційно-екстремальну інтелектуальну технологію (IEI-технологію), яка ґрунтується на максимізації інформаційної спроможності системи розпізнавання шляхом введення на етапі навчання додаткових інформаційних обмежень. Основні концептуальні положення IEI-технології такі:

- IEI-технологія ґрунтується на прямій оцінці інформаційної

спроможності системи, що навчається;

- IEI-технологія дозволяє оптимізувати просторово-часові параметри функціонування системи, що навчається;
- прийняття рішень у рамках IEI-технології здійснюється в рамках детерміновано-статистичного підходу шляхом побудови відносно простого детермінованого класифікатора, статистична корекція якого здійснюється в процесі навчання з метою побудови безпомилкових, за навчальними матрицями, вирішальних правил;
- методи IEI-технології ґрунтуються на застосуванні гіпотез як чіткої, так і нечіткої компактності реалізацій образу, тобто є працездатним за умови перетину класів розпізнавання, що має місце в практичних задачах контролю та керування;
- методи IEI-технології базуються на вибірковому підході математичної статистики і орієнтовані на застосування прийнятних з практичних міркувань обсягів репрезентативних навчальних вибірок;

Побудова «точного» контейнера класу розпізнавання складної геометричної форми у багатовимірному просторі ознак навіть для сучасних комп'ютерних комплексів має суттєві ускладнення. Тому при обґрунтованості гіпотези нечіткої компактності в працях [1, 71] згідно з принципом редукції у рамках детерміновано-статистичного підходу пропонується відновлювати контейнери спрощеної форми, наближеної до «точної» деяким оптимальним способом, і які формують вирішальні правила, що забезпечують в режимі екзамену достовірність прийняття рішень, близьку до максимальної асимптотичної.

Інформаційно-екстремальне машинне навчання реалізує процедуру адаптивного кодування ознак двійковим поданням шляхом пошуку оптимальних верхньої $A_{B,i}$ та нижньої $A_{H,i}$ меж поля контрольних допусків на значення i-ї ознаки. При цьому межі поля контрольних допусків для i-ї ознаки обчислюється за формулами:

$$A_{H,i} = \overline{y_{1,i}} - \delta_i,\ A_{B,i} = \overline{y_{1,i}} + \delta_i$$

де $\overline{y_{1,i}}$ – середнє вибіркове значення i-ї ознаки в базовому класі $X_1^o \in \{X_m^o\}$.

Формування бінарної навчальної матриці $\{x_{m,i}^{(j)} \mid \ i = \overline{1,N};\ j = \overline{1,n};\ m = \overline{1,M}\}$ здійснюється за правилами:

$$x_{m,i}^{(j)} = \begin{cases} 1, & \text{if} \quad A_{H,i} \le y_{m,i}^{(j)} \le A_{B,i}; \\ 0, & \text{if} \quad \text{else}; \end{cases}$$

Оптимізація параметра $\delta = \delta_i, i = \overline{1,N}$ поля контрольних допусків, здійснюється за ітераційною процедурою

$$\delta^* = \arg\max_{G_\delta} \left\{ \frac{1}{M} \sum_{m=1}^{M} \left[\max_{G_E \cap G_d} E_m \right] \right\}, \qquad (3.2.4)$$

де E_m – критерій ефективності навчання класифікатора розпізнавати реалізації m-го класу; G_δ – область допустимих значень параметра рецептивного поля; G_E – допустима область визначення функції інформаційного критерію; G_d – область допустимих значень геометричних параметрів контейнера класу X_m^o, що відновлюється в процесі машинного навчання в радіальному базисі простору ознак розпізнавання.

Вкладений в процедурі (3.2.4) алгоритм здійснює оптимізацію кодового радіусу d_m для гіперсферичного контейнера з еталонним (усередненим) бінарним вектором x_m для кожного класу розпізнавання. Розглянемо основні кроки цього алгоритму.

Крок 1. Формування масиву еталонних двійкових векторів-реалізацій $\{x_{m,i} \mid m=\overline{1,M},\ i=\overline{1,N}\}$, елементи яких визначаються за правилом

$$x_{m,i}=\begin{cases}1, & if \ \ \frac{1}{n}\sum_{j=1}^{n} x_{m,i}^{(j)} > 0{,}5;\\ 0, & if \ \ else,\end{cases}$$

Крок 2. Розбиття множини еталонних векторів на пари найближчих "сусідів": $\Re_m^{|2|}=\langle x_m, x_c\rangle$, де x_c – еталонний вектор сусіднього класу X_c^o, за такою схемою алгоритму:

1) структурується множина еталонних векторів, починаючи з вектора x_1 базового класу X_l^o, який характеризує найбільш бажаний функціональний стан;

2) будується матриця кодових відстаней між еталонними векторами розмірності $M \times M$;

3) для кожного рядка матриці кодових відстаней знаходиться мінімальний елемент, який належить стовпчику вектора, найближчого до вектора, що визначає рядок. За наявності декількох однакових мінімальних елементів вибирається з них будь-який, оскільки вони є рівноправними;

4) формується структурована множина елементів попарного розбиття $\{\Re_m^{|2|} \mid m=\overline{1,M}\}$, яка задає план навчання.

Крок 3. Оптимізація кодової відстані d_m відбувається за рекурентною процедурою

$$d_m(k)=[d_m(k-1)+h \mid d_m(k) \in G_m^d],$$

При цьому $E_m(0)=0$.

Крок 5. Процедура закінчується при знаходженні максимуму інформаційного критерію ефективності навчання в робочій області його визначення

$$E_m^* = \max_{\{d\}} E_m,$$

де $\{d\} = \{d_1,...,d_k,...,d_{\max}\} \in [0; d(x_m \oplus x_l) - 1]$ – множина радіусів концентрованих гіперсфер, центр яких визначається вершиною еталонного вектора $x_m \in X_m^o$.

Як критерій ефективності інформаційно-екстремального машинного навчання розглядається модифікація інформаційної міри Кульбака, в якій відношення правдоподібності представлено у вигляді відношення повної ймовірності правильного прийняття рішень P_{true} до повної ймовірності помилкового прийняття рішень P_{false}. Для випадку двох альтернативних і рівно ймовірних гіпотез міра Кульбака має вигляд

$$J_m = \left[P_{true,m} - P_{false,m}\right] \log_2 \frac{P_{true,m}}{P_{false,m}} =$$

$$= \left| \begin{array}{l} P_{true,m} = 0{,}5D_{1,m} + 0{,}5D_{2,m} \\ P_{false,m} = 0{,}5\alpha_m + 0{,}5\beta_m \\ \alpha_m = 1 - D_{1,m}; D_{2,m} = 1 - \beta_m \end{array} \right| =$$

$$= \left[D_{1,m} - \beta_m\right] * \log_2 \left(\frac{1 + \left[D_{1,m} - \beta_m\right]}{1 - \left[D_{1,m} - \beta_m\right]} \right), \qquad (3.2.5)$$

де $D_{1,m}$ – перша достовірність для класу X_m^o; $D_{2,m}$ – друга достовірність; α_m – помилка другого роду; β_m – помилка другого роду.

Для порівняльного аналізу результатів досліджень і при оцінці ступеню близькості реальної системи до потенційної доцільно використовувати нормовану модифікацію критерію (3.2.5) у вигляді

$$E_m = \frac{J_m}{J_{\max}}, \qquad (3.2.6)$$

де $J_{\max}$ – значення критерію при $D_{1,m} = 1$ і $\beta_m = 0$.

У режимі екзамену рішення про належність вектора-реалізації $x^{(j)}$ одному з класів алфавіту $\{X_m^o\}$ приймається шляхом обчислення геометричної функції належності

$$\mu_m = 1 - \frac{d(x_m^* \oplus x^{(j)})}{d_m^*}, \qquad (3.2.7)$$

де $d(x_m^* \oplus x^{(j)})$ – кодова відстань від еталонного вектора (центра контейнера) класу X_m^o до вектора $x^{(j)}$.

Визначення класу X_m^o, до якого належить реалізація образу, здійснюється за умови

$$\bar{\mu}_m^* = \max_{\{m\}} \bar{\mu}_m,$$

де $\bar{\mu}_m$ – усереднене значення функцій належності для реалізацій класу X_m^o, або видача повідомлення: «Клас не визначено», якщо $\bar{\mu}_m^* \le c$. Тут c – порогове значення.

При вирішенні задачі синтезу вирішальних правил системи керування інформаційно-телекомунікаційним середовищем на основі методів машинного навчання часто необхідно використовувати вибірки

даних великого розміру, що обумовлює потребу у використанні великих обсягів оперативної пам’яті та суттєво збільшує затрати машинного часу на обробку даних. Дані моніторингу компонентів інформаційно-телекомунікаційного середовища окрім великого обсягу, як правило, характеризуються значною незбалансованістю класів. В такій ситуації при використанні стандартних методів класифікації зменшення загальної помилки класифікації призводить до збільшення помилки класифікації реалізації класу аномального функціонального стану порівняно з мажоритарним класом нормального стану. Однією з простих технік врахування незбалансованості є збільшення “премій” за правильні класифікації реалізацій міноритарних класів. Такий підхід дозволяє підвищити чутливість до рідкісних подій, однак одночасно призводить до зменшення сумарної точності розпізнавання. У зв’язку з цим останнім часом набули популярності методи семплювання [72, 73], суть яких полягає у балансуванні обсягу вибірок класів шляхом генерації нових чи видалення дублючих (напівдублюючих) векторів-реалізації образів (рис.3.3).

Базовими методами семплювання є undersampling та oversampling, які можуть використовуватися як окремо, так і комбіновано в одній ітераційній процедурі формування навчальної вибірки. Застосування undersampling дозволяє збалансувати число реалізацій різних класів випадковою вибіркою з мажоритарного класу, однак це може призвести до втрати важливої інформації. Методи oversampling штучно збільшують кількість реалізацій міноритарного класу шляхом дублювання чи шляхом їх генерації за допомогою малих відхилень від реальних реалізацій міноритарного класу.

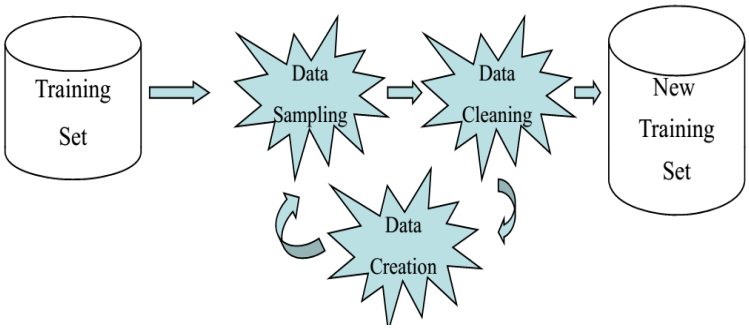


Рисунок 3.3 – Схема балансування класів методами семплювання

У задачах обробки великих даних, в тому числі незбалансованих, отримали поширення методи послідовної (boosting) та паралельної (bagging) композиції класифікаторів, що навчаються на різних, підвибірках [74].

В алгоритмі Bagging вхідні дані випадковим чином розбиваються на одинакові за розміром підмножини, кожна з яких використовується для навчання одного базового класифікатора (рис.3.4). При цьому кожен вектор-реалізація із вхідного набору даних має однакову ймовірність бути обраним для навчання базових класифікаторів.

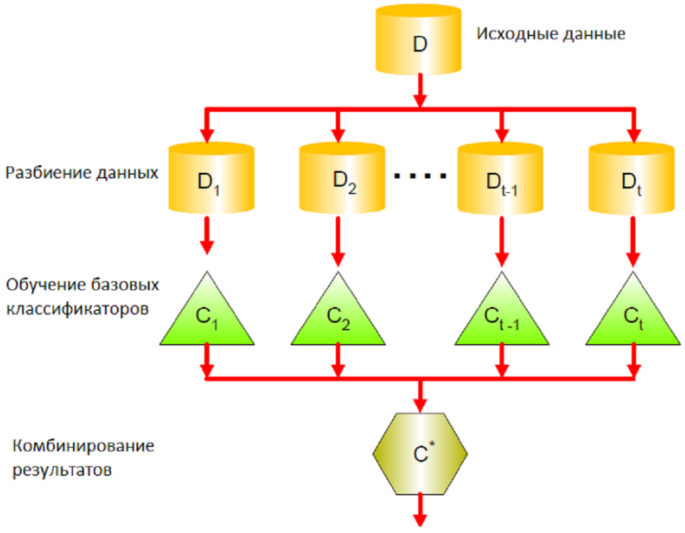


Рисунок 3.4 – Схема паралельної композиції класифікаторів

Прогноз паралельної композиції з K класифікаторів визначається більшістю голосів або їх усередненим значенням

$$a(x) = \max_{m=1..M} \sum_{k=1}^{K} count(a_k(x) = m),$$

або

$$a(x) = round\left(\frac{1}{K}\sum_{k=1}^{K} a_k(x)\right),$$

де $count(a_k(x) = m)$ – лічильник кількості правильного розпізнавання вхідного вектора x як реалізації класу X_m^o; $round()$ – функція округлення до найближчого цілого числа.

В алгоритмі Boosting базові класифікатори навчаються послідовно, а не паралельно, як в алгоритмі Bagging. Набір даних, на яких навчається кожен наступний базовий алгоритм в Boosting, залежить від точності прогнозування попереднього базового класифікатора (рис.3.5).

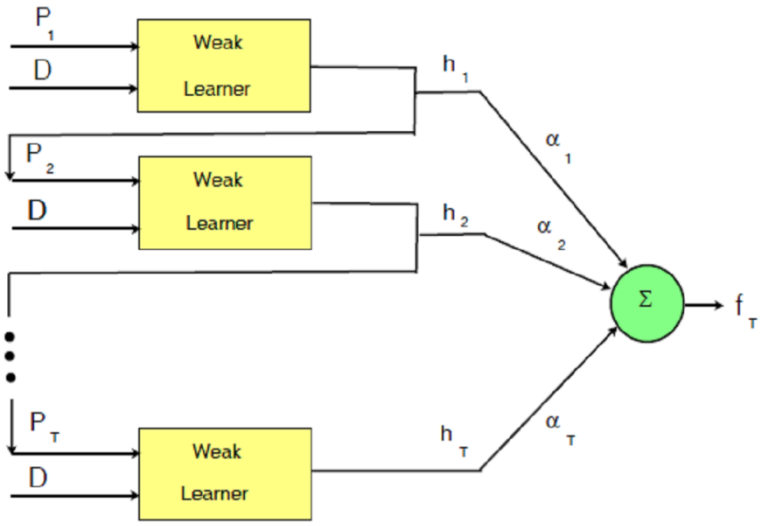


Рисунок 3.5 – Схема послідовної композиції класифікаторів

Важливим поняттям в Boosting є вага вектора-реалізації в наборі даних, яка оновлюється після кожної ітерації (виконання навчання базового класифікатора). Значення ваги вектора-реалізації описує її важливість для навчання наступного базового класифікатора і обчислюється на основі помилки прогнозу попереднього класифікатора

на цьому векторі-реалізації. Вагу вектора-реалізації можна інтерпретувати як ймовірність її вибору для навчання наступного класифікатора. Прогноз послідовної композиції класифікаторів здійснюється шляхом зваженого голосування.

$$a(x) = round\left(\sum_{k=1}^{K} \omega_k a_k(x) \right),$$

де ω_k – нормований ваговий коефіцієнт k-го базового класифікатора, для якого виконується умова

$$\sum_{k=1}^{K} \omega_k = 1$$

Розмір отримуваної композиції класифікаторів залежить від використовуваного методу навчання та особливостей вхідних даних.

В основу багатьох методів класифікації покладено гіпотезу компактності, яка передбачає, що об’єкти, які належать до певних класів, формують згустки точок – кластери в просторі ознак. Але в багатьох випадках функціонування класифікатора відбувається за умов невизначеності, що виникає через перекриття границь класів внаслідок складної конфігурації багатомодального розподілу векторів-реалізацій класів в просторі ознак (рис.3.6).

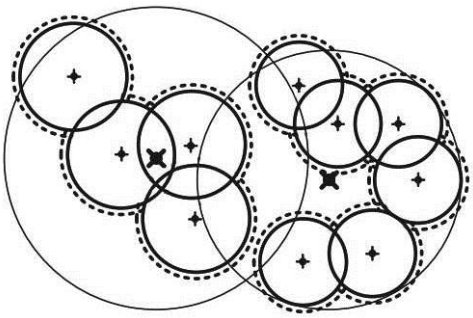


Рисунок 3.6 – Перетин гіперсферичних роздільних гіперповерхонь двох багатомодальних класів

Як показано на рис.3.6 для покращення точності класифікації в умовах нерівномірного та багатомодального розподілу навчальних векторів-реалізацій в класах варто розглянути представлення кожного класу декількома кластерами.

Тому для автоматизації формування вхідного математичного опису систем класифікаційного керування набуло поширення використання ідей і методів кластер-аналізу архівних даних моніторингу. В рамках цього підходу вектори-реалізації, що утворюють один кластер, можуть бути віднесені до одного з класів заданого алфавіту за результатами експертного аналізу лише одного прецеденту з даного кластеру.

В основі кластер-аналізу лежить гіпотеза компактності. Тобто апріорно некласифікована навчальна вибірка у просторі ознак складається із множини згущень реалізацій образів, які мають відповідні центри їх розсіювання. Реалізації, що відносяться до одного згущення, яке будемо називати таксоном, є найближчими до еталонного вектора-реалізації, який визначає центр розсіювання реалізації і є найближчими відносно векторів інших таксонів. Класичним представником таких методів є алгоритм k-середніх. Розглянемо основні кроки цього алгоритму.

Крок1. Дослідник задає кількість кластерів k, що необхідно утворити.

Крок 2. Випадковим чином із вибірки обирається *k* векторів-реалізацій, які на цьому кроці вважаються центрами кластерів.

Крок 3. Кожний вектор-реалізація «приписується» до одного з *k* кластерів, відстань до якого найкоротша.

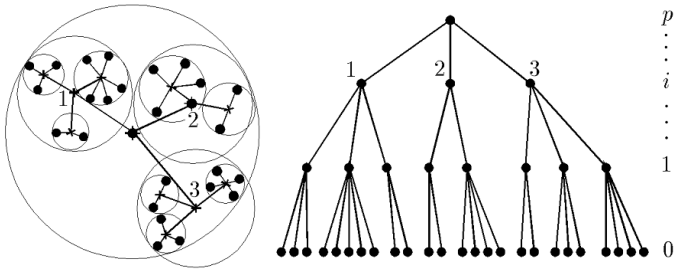
Крок 4. Розраховується новий центр кожного кластера як елемент, ознаки якого розраховуються як середнє арифметичне ознак об'єктів, що входять у цей кластер.

Крок 5. Ітераційне повторення кроків 3-4 до стабілізації складу кластерів.

Алгоритм *k*-середніх є прикладом ітераційних методів кластер-аналізу, у яких оптимальна кількість кластерів визначається шляхом повторення кроків алгоритму 2-5 починаючи при різних значеннях *k*, порівнюючи отримані результати. Найкращим розбиттям вважається те, яке забезпечує мінімальні внутрішньокластерні відстані (дисперсію всередині кластерів) та максимальну міжкластерну відстань (відстань між центрами кластерів).

Ще одним популярним ітераційного алгоритмом кластер-аналізу є FOREL, де використовуються кластери гіперсферичної форми, а їх кількість залежить від радіуса гіперсфери. На початку алгоритму будується гіперсфера мінімального радіуса, яка охоплює всі об’єкти вибірки. Кожне зменшення радіусу супроводжується обчисленням центру ваги векторів-реалізацій, що знаходяться в межах гіперсфери, і переносом центру гіперсфери в отриманий центр ваги. Умовою зупину процедури центрування є стабілізація складу внутрішніх об’єктів кластера і незмінність координат центру. Внутрішні об’єкти кластера вилучаються з розгляду при побудові гіперсфери наступного кластера.

Алгоритми *k*-середніх і FOREL чутливі до вибору початкової точки пошуку. З метою наглядного і детального подання структури даних використовують підхід ієрархічного кластер-аналізу, який не потребує визначення кількості кластерів, а здійснює побудову повного дерева вкладених кластерів (дендрограми) (рис.3.7) [75].



а *б*

Рисунок 3.7 – Схема ієрархічного кластер-аналізу: *а* – геометричне розбиття простору ознак; *б* – дендрограма

Серед алгоритмів ієрархічної класифікації виділяють два основні типи: висхідні (агломеративні) і спадні (дивізимні) алгоритми. В агломеративних алгоритмах на початку роботи кожний вектор-реалізацію розміщують в окремому кластері, а потім об’єднують кластери в більші, доки всі вектори-реалізації розподілу не будуть міститися в одному кластері. Дивізимні алгоритми працюють за принципом «зверху-вниз»: на початку всі реалізації розміщують в один клас, який потім розбивають на менші кластери.

Вектори-реалізації функціонального стану компонентів та сервісів інформаційно-телекомунікаційного середовища, як правило мають високу розмірність. До словника ознак можуть входити дані про трафік, канали зв’язку, мережеві пристрої та обчислювальні ресурси вузлів обробки. Такі дані збираються стандартними системами керування з різноманітних сенсорів та отримуються від програмного забезпечення мережевого рівня на клієнтському та мережевому обладнанні. Однак для вирішення поточних задач керування доступністю, конфігураціями, продуктивністю та безпекою ІТ-інфраструктури корисною є лише частина доступної інформації. Тому для підвищення оперативності машинного навчання та достовірності вирішальних правил можуть бути застосовані різні методи редукції словника ознак.

У загальному випадку процес редукції словника ознак є ітераційним, кожна ітерація якого включає етапи генерації варіантів словника, оцінки їх ефективності та перевірки критерію зупину [11, 76, 77]. Етап генерації словників може починатися як з варіанту словника, що включає всі ознаки, так і з пустого або випадково сформованого. Наступний словник створюється шляхом додавання/видалення ознак із сформованого на попередній ітерації варіанту або повторного випадкового формування. Оцінка ефективності словника ознак може здійснюватися за дистанційними, інформаційними, кореляційними, сумісними та точнісними мірами якості. Умовами зупину можуть бути

досягнення граничного значення міри якості, граничної кількості ітерацій чи відсутність зростання міри якості після виконання ітерації алгоритму.

Таким чином, дані, що були накопичені в процесі моніторингу процесу функціонування інформаційно-телекомунікаційного середовища, можуть бути використані для формування за допомогою алгоритмів машинного навчання та кластер-аналізу оптимальних в інформаційному сенсі вирішальних правил для передбачення ситуацій, які призводять до матеріальних втрат чи втраченого прибутку.

3.3 Алгоритми оптимізації параметрів функціонування системи керування інформаційно-телекомунікаційним середовищем

Зведення частинних критеріїв ефективності навчання системи керування інформаційно-телекомунікаційним середовищем до єдиного інтегрального критерію J дозволяє перейти до розгляду однокритеріальної задачі багатопараметричної оптимізації. При цьому вектор просторово-часових параметрів функціонування системи керування можна подати в такому вигляді:

$$g = < g_1,...,g_{\Xi_1},\ f_1,...,f_{\Xi_2},\ \psi_1,...,\psi_{\Xi_3} >, \qquad (3.3.1)$$

де $< g_1,...,g_{\xi_1},...,g_{\Xi_1} >$ – параметри, які впливають на вартість експлуатації і обслуговування системи керування; $< f_1,...,f_{\xi_2},...,f_{\Xi_2} >$ – параметри, які впливають на кодування контрольованих характеристик; $< \psi_1,...,\psi_{\xi_3},...,\psi_{\Xi_3} >$ – параметри, які впливають на кодування функціональних станів.

Обмеження на відповідні параметри функціонування можуть бути задані у вигляді систем нерівностей:

$$R_{\xi_1}(g_1,...,g_{\xi_1},...,g_{\Xi_1}) \le 0;$$

$$R_{\xi_2}(f_1,...,f_{\xi_2},...,f_{\Xi_2}) \le 0;$$

$$R_{\xi_3}(\psi_1,...,\psi_{\xi_3},...,\psi_{\Xi_3}) \le 0.$$

Згідно з принципом ієрархічної впорядкованості та методологією об'єктно-орієнтованого проектування процес синтезу прогностичних вирішальних правил системи керування можна представити як ієрархічну ітераційну процедуру оптимізації структурованих параметрів (3.3.1)

$$g^* = <\arg\left\{\max_G\left\{\max_F\left\{\max_\Psi J\right\}\right\}\right\}>,$$

де G – області допустимих значень множи параметрів $< g_1,...,g_{\xi_1},...,g_{\Xi_1} >$; F – області допустимих значень множини параметрів $< f_1,...,f_{\xi_2},...,f_{\Xi_2} >$; Ψ – області допустимих значень множини параметрів $< \psi_1,...,\psi_{\xi_3},...,\psi_{\Xi_3} >$.

В робочому режимі система керування повинна прийняти оптимальне оперативне рішення s^* щодо розподілу ресурсів інформаційно-телекомунікаційного середовища. Це рішення знаходиться в процесі мінімізації комплексного критерію (3.1.9), що оцінює рівень енергоспоживання та прогнозованих порушень SLA-угод

$$s^* = \arg\min_{s\in S} \sum_{i=1}^{I} \lambda_i k_i(s),$$

де S – область допустимих рішень; $k_i(s)$ – нормалізовані значення частинних критеріїв ефективності рішення $s \in S$ щодо розподілу ресурсів; λ_i – коефіцієнти важливості (пріоритетності) частинних критеріїв.

Кожне рішення s щодо розподілу ресурсів за визначеною схемою кодується вектором дійсних або двійкових чисел

$$s = < s_1, \ldots, s_r, \ldots, s_R > \qquad (3.3.2)$$

Вибір схеми кодування рішення s залежить як від виду ресурсів і типу задач, що їх споживають, так і вибору частинних критеріїв. Для розподілу обчислювальних задач між віртуальними обчислювальними вузлами вектор s може складатися з числових значень моментів запуску множини задач на заданій множині вузлів. Крім того розподіл задач можна задати у вигляді бітового рядка, що представляє двійкову матрицю впорядкування завдань, де елементи одного її рядка представляють послідовність виконання задач на одному обчислювальному вузлі. Для розподілу мережевих ресурсів між потоками пакетів, що надходять на порти маршрутизатора, вектор s може складатися з вагових коефіцієнтів дистанційних метрик для кожного каналу зв'язку.

Таким чином задача синтезу прогностичних вирішальних правил та задача планування використання ресурсів інформаційно-телекомунікаційної системи зводяться до задачі однокритеріальної багатопараметричної оптимізації. При цьому гетерогенність, ієрархічність інформаційно-телекомунікаційного середовища, багатофакторність та нестаціонарність процесів споживання ресурсів обумовлюють нелінійність, багатоекстремальність та високу розмірність оптимізаційної задачі. Для ефективного вирішення оптимізаційних задач з подібними характеристиками у 80-х роках минулого століття почали інтенсивно розроблятися стохастичні пошукові алгоритми оптимізації. Останнім часом в даному напрямку спостерігається стрімке зростання інтересу до розробки алгоритмів, в основі яких лежать ідеї, запозичені в природи, а також базові постулати універсальності, фундаментальності, властиві самоорганізації природних систем. Одним з таких напрямків є мультиагентні методи інтелектуальної оптимізації, що базуються на

моделюванні колективного (ройового) інтелекту (Swarm intelligence) [26, 78].

Алгоритми ройового інтелекту відносяться до класу евристичних алгоритмів. Евристичний алгоритм – це алгоритм, призначений для більш швидкого вирішення проблеми в порівнянні з класичними методами, або для знаходження наближеного ефективного рішення, коли інші методи не в змозі знайти точне рішення. Це досягається за рахунок оптимальності, повноти, точності або швидкості виконання алгоритму.

В рамках «колективного» інтелекту глобальна поведінка всієї системи розглядається, як результат взаємодій ряду простих агентів. Прихильники напряму «інтелекту рою» спираються на такі положення:

1) багатоагентна система – це популяція простих і залежних один від одного агентів;

2) кожен агент самостійно визначає свої реакції на події в локальному середовищі та взаємодії з іншими агентами;

3) зв'язки між агентами є горизонтальними, тобто не існує агента-супервізора, що керує взаємодією інших агентів;

4) немає точних правил, що дозволяють визначити глобальну поведінку агентів;

5) поведінка, властивості і структура на колективному рівні породжуються тільки локальними взаємодіями агентів.

В даний час розроблено і з успіхом використовуються на практиці безліч алгоритмів, основаних на інтелекті рою: алгоритм мурашиних колоній, алгоритм рою бджіл, алгоритм косяків риб, алгоритм летючих мишей, алгоритм зграї птахів, алгоритм рою частинок та інші. Загальна схема ройових алгоритмів включає наступні етапи:

1) ініціалізація рою шляхом створення, як правило, випадковим чином в просторі пошуку деякої кількості початкових наближень до шуканого рішення;

2) міграція агентів рою шляхом застосування міграційних операторів (специфічних для кожного ройового алгоритму), які переміщують агентів в області пошуку, наближаючи їх до екстремуму критерію оптимізації;

3) перевірка умови зупину алгоритму, невиконання якої означає повернення до другого етапу.

Умовою зупину ройового алгоритму оптимізації може бути або досягнення заданого числа ітерацій, або стан стагнації, коли краще досягнуте значення критерію оптимізації не змінюється в продовж заданого числа ітерацій.

Розглянемо один з найбільш популярних алгоритмів ройової оптимізації – алгоритм рою частинок (Particle Swarm Optimization, PSO), оскільки він дозволяє знайти глобальний максимум критерію, не потребує початкових наближень і відрізняється простотою реалізації [78]. Завдяки випадковості розподілу частинок і їх хаотичності в русі з'являється дуже велика імовірність знайти оптимальне рішення за декілька ітерацій. Елемент випадковості в процесі пошуку забезпечується параметрами алгоритму, значення яких генеруються випадковим чином з заданого діапазону $(0,1)$ у відповідності з нормальним законом розподілу $U(0,1)$.

Ефективність кожної частинки, тобто її близькість до глобального оптимуму, вимірюється за допомогою наперед визначеної фітнес-функції, яка використовується як критерій оптимізації. При цьому кожна частинка зберігає наступну інформацію: P_j – поточна позиція j-ї частинки; V_j – поточна швидкість частинки, $Pbest_j$ – краща персональна позиція частинки. Краща персональна позиція j-ї частинки – це позиція j-ї частинки, в якій значення фітнес функції для частинки було максимальним на поточний момент часу. Крім цього, з метою пошуку глобального екстремуму фітнес-функції найкраща частинка шукається в усьому рої, а її позиція позначаться як $Gbest$. Розглянемо основні кроки

реалізації алгоритму рою частинок для оптимізації вектора параметрів рішення (3.3.2).

1. Ініціалізація рою:

1) ініціалізація кількості частинок ;

2) ініціалізація розмірності кожної частинки R та ініціалізація меж зміни параметра s_r ;

3) ініціалізація початкових позицій частинок

$$P_j(0) := s_{\max} U(0,1);$$

4) ініціалізація початкових швидкостей частинок $V_j(0) := 0$;

5) ініціалізація максимальної швидкості частинок в $V_{\max,r}$;

6) ініціалізація вагових коефіцієнтів для формули швидкості, тобто ваги інерції w та констант прискорення c_1 і c_2.

2. Інкремент номера ітерації: $k := k+1$.

3. Інкремент номера частинки: $j := j+1$.

4. Інкремент номера координати в позиції: $r := r+1$.

5. Розрахунок нового стану частинки:

1) розрахунок r-ї компоненти швидкості для j-ї частинки за правилами

$$V_{j,r}(k+1) := wV_{j,r}(k) + c_1 a_{1,r}(k) * [Pbest_{j,r}(k) - P_{j,r}(k)] + c_2 a_2(k) * [Gbest_j - P_{j,r}(k)],$$

$$V_{j,r}(k+1) := \begin{cases} V_{j,r}(k+1) & якщо \quad V_{j,r}(k+1) < V_{\max,r}, \\ V_{\max,r} & якщо \quad V_{j,r}(k+1) \ge V_{\max,r}, \end{cases}$$

де $a_1(k) = U(0,1),\ a_2(k) = U(0,1);$

2) оновлення позиції частинки

$$P_j(k+1) := P_j(k) + V_j(k+1);$$

3) перевірка перетину меж простору, і зміна швидкості у випадку перетину за правилом

$$V_{j,r}(k+1) := V_{j,r}(k+1)\alpha,$$

де α – параметр типу межі простору ($\alpha = 0$ – поглинаюча межа; $\alpha = 1$ – прозора; $\alpha = -1$ – відзеркалююча; $\alpha = -U(0,1)$ – демпфуюча);

4) обчислення цільової функції $J_j(k+1)$;

5) оновлення значень найкращої персональної $Pbest$ та глобальної $Gbest$ позицій

$$Pbest_j(k+1) := \begin{cases} Pbest_j(k), якщо J(P_j(k+1)) \le J(Pbest_j(k)); \\ P_j(k+1), якщо інакше; \end{cases}$$

$$Gbest(k+1) := \arg\max_j \{J(Pbest_j(k+1))\}.$$

6. Перевірка умови зупину: якщо

$$(k < k_\max) \wedge (J(Gbest(k+1) < J_\max),$$

де $k_\max$ – задана максимальна кількість ітерацій; $J_\max$ – граничне значення фітнес-функції, то перехід до кроку 2, інакше – до кроку 7.

7. ЗУПИН.

Ще одним популярним представником ройових алгоритмів оптимізації є алгоритм пошуку косяком риб (Fish School Search, FSS), який відрізняється простотою реалізації, інтерпретабельністю та високою швидкістю збіжності.

В алгоритмі FSS кожна риба зберігає одне з рішень задачі. При цьому косяк риб є агрегацією агентів рою, які рухаються приблизно з однією й тією ж швидкістю і орієнтацією, підтримуючи приблизно однакову відстань між собою. Індивідуальний успіх кожної риби в пошуці рішення характеризується її вагою, що відіграє роль пам'яті. Кожна ітерація пошуку виконує дві групи операторів – оператори годування та оператори плавання.

Оператор годування формалізує успішність дослідження агентами тих чи інших областей "акваріуму" і полягає в обчисленні ваги z-го агента, яка пропорційна нормалізованій різниці значень фітнес-функції на наступній та поточній ітераціях

$$w_z[k+1] = w_z[k] + \frac{J(P_z[k+1]) - J(P_z[k])}{\max(J(P_z[k+1]), J(P_z[k]))},\ z = \overline{1, Z},$$

де $P_z[k+1]$, $P_z[k]$ – позиція z-го агента в багатовимірному просторі рішень на k-й і $(k+1)$-й ітераціях алгоритму FSS.

Максимально можливе значення ваги агента w_z у алгоритмі FSS обмежується значенням $w_{\max} > 0.$ При цьому під час ініціалізації популяції всім агентам присвоюється вага, що рівна $w_{\max} \cdot 0{,}5.$

В алгоритмі FSS розрізняють три види плавання – індивідуальне, інстинктивно-колективне та колективно-вольове. Ці види плавання здійснюються послідовно один за одним в окремі інтервали часу $(t, \tau], (\tau, \theta], (\theta, t^{'}),\ t < \tau < \theta < t^{'}, t^{'} = t + 1.$

Під час індивідуального плавання агентів відбувається їх переміщення, що має рівноймовірний випадковий характер. При цьому за одну ітерацію алгоритму FSS крок індивідуального плавання виконується фіксовану кількість разів. Компоненти кроку переміщення V_z^{ind} рівномірно розподілені в заданому інтервалі $v_{\max}^{ind}$

$$V_z^{ind} = U(0;1)v_{\max}^{ind},\ z = \overline{1,Z},$$

де $U(0;1)$ – випадкове число із заданого діапазону $(0;1)$.

В процесі інстинктивно-колективного плавання на кожного з агентів чинять вплив всі інші агенти популяції і цей вплив пропорційний індивідуальним успіхам агентів. При цьому позиції агентів обчислюються за формулою

$$P_z^{\theta} = P_z^{\tau} + \frac{\sum_j V_j^{ind}(\tau)(J(P_j^{\tau}) - J(P_j^{t}))}{\sum_j J(P_j^{\tau}) - J(P_j^{t})},\ z = \overline{1,Z}$$

Колективно-вольове плавання полягає у зміщенні всіх агентів у напрямку поточного центру тяжіння популяції за умови збільшення сумарної ваги косяка риб в результаті індивідуального та інстинктивно-колективного плавання. Якщо сумарна вага зменшилась, то зміщення відбувається в протилежному напрямку. Колективно-вольове плавання виконується за правилами

$$P_z^{'} = P_z^{\theta} \pm v^{vol}(P_z^{\theta} - P_c^{\theta}),\ z = \overline{1,Z}, \qquad (3.3.3)$$

де P_c^{θ} – координати центра тяжіння косяка риб, що визначаються за формулою

$$P_c^{\theta} = \frac{\sum_z w_z^{\theta} P_z^{\theta}}{\sum_z w_z^{\theta}}.$$

У формулі (3.3.3) знак плюс використовується за умови

$$\sum_z w_z^{\theta} > \sum_z w_z^{\theta-1},$$

а знак мінус – в протилежному випадку.

При цьому розмір кроку переміщення агентів v^{vol} є випадкою величиною

$$v^{vol} = v_{\max}^{vol} U(0;1),$$

де $v_{\max}^{vol}$ - максимально допустима довжина кроку переміщення при колективно-вольовому плаванні.

Однією з основних проблем конструювання ройових алгоритмів є проблема забезпечення балансу між швидкістю збіжності алгоритму, яка характеризує швидкість зменшення різномантіності в рою, і диверсифікацією пошуку, яка характеризує широту пошуку і відповідає за збереження різноманітності в рою. Найбільш розвиненими механізмами вирішення проблеми балансу є адаптація та самоадаптація ройових алгоритмів. Реалізація цих механізмів передбачає використання допоміжного фонового алгоритму пошуку, що здійснює настройку параметрів базового ройового алгоритму і забезпечує максимально можливу швидкість його збіжності як це показано в праці [78].

Таким чином, синтез прогностичних вирішальних правил системи керування інформаційно-телекомунікаційним середовищем та безпосереднє прийняття рішень передбачають пошук екстремуму функції критерію багатьох параметрів. При цьому одним з найбільш ефективних підходів до пошуку глобального оптимуму оптимізаційної задачі даного типу є використання ідей і методів ройового інтелекту.

3.4 Інформаційно-екстремальний метод ідентифікації типу мережевого трафіку

Сучасні телекомунікаційні мережі характеризуються високими і надвисокими показниками пакетообігу, що обумовлює необхідність пріоритезації трафіку відповідно до вимог рівня обслуговування користувачів – Service Level Agreement (SLA) та якості мережевих сервісів – Quality of Service (QoS). Для визначення пріоритету при формуванні смуги пропускання окремого трафіку необхідно мати інструмент точної ідентифікації та класифікації трафіку. При цьому поширення мережевих додатків, які динамічно змінюють порти транспортних протоколів, використовують шифрування та інкапсуляцію трафіку в тунельний протокол, призводить до низької ефективності класифікації трафіку на основі портів чи корисного навантаження (Deep Packet Inspection). Вирішення цих проблем пов'язується з використанням методів машинного навчання та розпізнавання образів, де, як правило, ознаками розпізнавання виступають статистичні характеристики потоку пакетів (Flow-based Classification) [60, 65, 72]. Проте різноманітність та параметрична невизначеність інформаційних процесів, які генерують мережевий трафік, використання методів маскування трафіку призводять до значного перетину класів розпізнавання в просторі ознак, що зменшує достовірність класифікації [72, 79]. Особливо критично це проявляється при збільшенні потужності алфавіту класів та незбалансованості навчальних наборів даних [62, 72].

Одним із перспективних шляхів підвищення точності класифікації мережевого трафіку є інформаційний синтез класифікатора в рамках інформаційно-екстремальної інтелектуальної технології (ІЕІ-технологія), яка дозволяє трансформувати апріорно нечітке розбиття простору ознак в чітку еквівалентність класів розпізнавання. Крім цього, важливими перевагами ІЕІ-технології, є невисока обчислювальна складність

вирішальних правил, що сприяє економії ресурсів мережевого пристрою в умовах виского пакетообігу.

Розглянемо на прикладі додатків VoIP, SSH і BitTorrent задачу інформаційного синтезу здатної навчатися системи ідентифікації зашифрованого мережевого трафіку з ієрархічною структурою вирішальних правил, побудованих за незбалансованими наборами даних.

Нехай дано алфавіт $\{X_m^o \mid m=\overline{1,M}\}$ класів розпізнавання, які характеризують трафік різних типів мережевого сервісу. Як реалізації кожного класу розпізнавання розглядаються впорядковані вектори статистичних характеристик потоку пакетів. Відома навчальна матриця $\{y_{m,i}^{(j)} \mid m=\overline{1,M};\ j=\overline{1,n_m};\quad i=\overline{1,N}\}$, де M – потужність алфавіту класів розпізнавання; n_m – кількість реалізацій класу X_m^o; N – кількість ознак розпізнавання.

Відомі s-ярусна деревоподібна бінарна ієрархічна структура класів розпізнавання і структурований вектор параметрів

$$g = < \delta_{s_i}, d_s >, s = \overline{1, \mathrm{M}-1} \qquad (3.4.1)$$

де $\delta_{i,s}$ – параметр, який визначає півширину симетричного рецептивного поля для i-ї ознаки розпізнавання відносно усередненого значенню i-ї ознаки класу $X_{\mathrm{s}}^o \in \{X_{\mathrm{m}}^o\}$, який відокремлюється у листок дерева на s-му ярусі; d_s – кодовий радіус гіперсферичного (вписаного в одиничний гіперпаралелепіпед) контейнера класу X_s^o, який відновлюється в радіальному базисі бінарного простору Хеммінга $\Omega_Б$ з центром, що визначається вершиною одиничного еталонного вектора $x_{s,i}=1,\ i=\overline{1,N}$.

При цьому відомі обмеження на параметри функціонування вектора

(3.4.1): $\delta_{s_i} \in [0; 0{,}5 \cdot \delta_{H,i}]$, де $\delta_{H,i}$ – максимальна ширина рецептивного поля для i-ї ознаки розпізнавання; $0 < d_s < N$.

Необхідно в процесі навчання класифікатора трафіку визначити оптимальні значення координат вектора параметрів функціонування (3.4.1), які забезпечують максимальне значення усередненого за алфавітом класів розпізнавання критерію функціональної ефективності (КФЕ)

$$\overline{E} = \frac{1}{M-1} \sum_{s=1}^{M-1} E_s \ , \qquad (3.4.2)$$

де E_s – інформаційний критерій функціональної ефективності навчання класифікатора розпізнавати реалізації класу X_s^o; $\{k\}$– упорядкована множина кроків навчання (відновлення контейнерів класів розпізнавання).

При функціонуванні класифікатора трафіку в режимі розпізнавання необхідно прийняти рішення про належність реалізації мережевого трафіку до одного з класів мережевих сервісів $\left\{X_m^o \mid m = \overline{1, M}\right\}$ і призначити трафіку клас обслуговування, що відповідає вимогам SLA.

Для формування деревоподібної бінарної ієрархічної структури вирішальних правил послідовно на кожному ярусі здійснюється розбиття більших груп класів на дві менші. У найпростішому випадку достатньо здійснювати на кожному ярусі побудову контейнера для одного класу розпізнавання для його відокремлення в листок дерева. При цьому рецептивні поля для ознак розпізнавання визначаються на кожному ярусі окремо.

Квазіоптимізація параметра $\delta_s = \delta_{s_i}, i = \overline{1, N}$ рецептивного поля призначена для визначення стартових значень, які відповідають робочій області визначення функції інформаційного КФЕ, і здійснюється за ітераційною процедурою

$$\delta_s^* = \arg\max_{G_\delta}\left\{\frac{1}{M-1}\sum_{s=1}^{M-1}\left[\max_{G_E \cap G_d} E_s\right]\right\}, \qquad (3.4.3)$$

де E_s – КФЕ навчання класифікатора на s-ярусі ієрархічної структури G_δ – область допустимих значень параметра рецептивного поля; G_d – область допустимих значень радіусів гіперсферичних контейнерів класів розпізнавання; G_E – робоча область визначення функції КФЕ.

Послідовна оптимізація параметра δ_{s_i} рецептивного поля для i-ї ознаки здійснюється за ітераційною процедурою

$$\delta_{s_i}^* = \arg\left\{\bigotimes_{l=1}^{L}\max_{G_\delta}\left\{\frac{1}{M-1}\sum_{m=1}^{M-1}\left[\max_{G_E \cap G_d} E_s^{(l)}\right]\right\}\right\}, \qquad (3.4.4)$$

де $E_s^{(l)}$ – КФЕ навчання класифікатора на s-ярусі при l-му прогоні послідовної процедури оптимізації; $\otimes$ – символ операції повторення; L – кількість прогонів ітераційної процедури послідовної оптимізації контрольних допусків.

Базовий алгоритм [6,8] є вкладеним в процедури (3) та (4) і здійснює побудову контейнера тільки для базового класу $\mathrm{X}_\mathrm{s}^\mathrm{o} \in \{\mathrm{X}_\mathrm{m}^\mathrm{o}\}$, що відокремлюється у листок деревоподібної структури

$$d_s^* = \arg\max_{\{d_s\} \in G_d} E_s, \qquad (3.4.5)$$

Вибір базового класу $X_s^o \in \{X_m^o\}$ для s-го ярусу ієрархічної структури базується на ідеї найкращого відокремлення від реалізацій

чужих класів і здійснюється за таким алгоритмом LEARNING-1:

Крок 1. Ініціалізація лічильника ярусів дерева рішень: $s := 0$.

Крок 2. Ініціалізація лічильника класів: $k := 1$.

Крок 3. Оптимізація радіусу гіперсферичного контейнера класу X_k^o за ітеративною процедурою (3.4.5), прийнявши за сусідній клас X_c^o сукупність найближчих до ядра класу X_k^o реалізацій

$$\{x_c^{(j)} \mid j = \overline{1, n_k}\} \in \left[\bigcup_{c=1}^{M-h} X_c^o \right] \setminus X_k^o .$$

Крок 4. $k := k + 1$.

Крок 5. Порівняння: якщо $k < M - s$, то виконується крок 3, інакше – крок 6.

Крок 6. Прийняти за листок дерева клас, що забезпечує максимальне значення інформаційного КФЕ

$$X_s^o = \arg\{\max_{\{X_m\}}\{\max_{G_\delta}\{\max_{G_E}\{E_k\}\}\}$$

і вилучити його з подальшого розгляду.

Крок 7. $s := s + 1$.

Крок 8. Якщо $s < M - 1$, то перехід на крок 2, інакше – «ЗУПИН».

Як КФЕ навчання класифікатора на s-му ярусі розглянемо модифіковану інформаційну міру Кульбака [7,8], в якій відношення правдоподібності представлено у вигляді відношення повної ймовірності правильного прийняття рішень P_{true} до повної ймовірності помилкового прийняття рішень P_{false}. В цьому випадку для двохальтернативних гіпотез міра Кульбака має вигляд

$$E_s^{(k)} = \left[P_{true,s}^{(k)} - P_{false,s}^{(k)}\right] \log_2 \frac{P_{true,s}^{(k)}}{P_{false,s}^{(k)}} = = \left| \begin{array}{l} P_{true,s}^{(k)} = p_1 D_{1,s} + p_2 D_{2,s} \\ P_{false,s}^{(k)} = p_1 \alpha_s + p_2 \beta_s \\ p_1 = \dfrac{n_s}{n_s + n_c};\ p_2 = \dfrac{n_c}{n_s + n_c} \\ \alpha_s = 1 - D_{1,s};\ D_{2,s} = 1 - \beta_s \end{array} \right| =$$

$$= \frac{\left[n_c - n_s + 2 \cdot (n_s D_{1,s}^{(k)} - n_c \beta_s{}^{(k)})\right]}{n_s + n_c} * \log_2 \left(\frac{n_c + (n_s D_{1,s}^{(k)} - n_c \beta_s{}^{(k)})}{n_s - (n_s D_{1,s}^{(k)} - n_c \beta_s{}^{(k)})} \right), \quad (3.4.6)$$

де $D_{1,s}^{(k)}$ – перша достовірність, обчислена на k-му кроці навчання для s-го ярусу; $D_{2,s}^{(k)}$ – друга достовірність; $\alpha_s{}^{(k)}$ – помилка першого роду; $\beta_s{}^{(k)}$ – помилка другого роду; n_s – кількість реалізацій у навчальній вибірці базового класу X_s^o; n_c – кількість сусідніх реалізацій, що належать до інших класів s-го ярусу.

Нормовану модифікацію критерію (3.4.6) представимо у вигляді

$$\hat{E}_s^{(k)} = \frac{E_s^{(k)}}{E_{\max}}, \quad (3.4.7)$$

де $E_{\max}$ – значення критерію при $D_{1,h}^{(k)} = 1$ і $\beta_h{}^{(k)} = 0$.

При цьому робоча (допустима) область визначення функції інформаційного КФЕ обмежена нерівностями $D_1 \geq 0{,}5$ і $D_2 \geq 0{,}5$.

Визначення належності тестової реалізації $x^{(j)}$ до контейнера класу X_s^o здійснюється за правилом

$$if\ \ d[x_s \oplus x^{(j)}] \leq d_s\ \ then\ \ x^{(j)} \in X_s^o\ else\ \ x^{(j)} \notin X_s^o, \quad (3.4.8)$$

де $d[x_s \oplus x^{(j)}]$ – кодова відстань між векторами $x^{(j)}$ і x_s; d_s – радіус контейнера класу X_s^o, що відновлюється в бінарному просторі

ознак на s-му ярусі .

Як правило дані архівів моніторингу мережевого трафіку мають дуже великий обсяг та характеризуються різноманітністю і незбалансованістю, різним розподілом різних типів трафіку. Навчання класифікатора з використанням повного обсягу навчальних даних займе досить тривалий час, а формування вибірок меншого обсягу не гарантуватиме їх репрезентативності і призведе до втрат інформації. Для підвищення оперативності навчання інформаційно-ектремального класифікатора трафіку запропонуємо наступну модифікацію алгоритму навчання LEARNING-2:

Крок 1. Ініціалізація масивів оптимального $\delta^*_{s_i}$ та стартового $\delta^{start}_{s_i}$ параметрів рецептивних полів для ознак розпізнавання: $\delta^*_{s_i} := 0$; $\delta^{start}_{s_i} := 0$ при $i = \overline{1, N}$.

Крок 2. Поділ великого масиву апріорно-класифікованих векторів-реалізацій на навчальну $Y = \{y^{(j)}_{m,i} \mid m = \overline{1, M}; j = \overline{1, n_{\min}}; i = \overline{1, N}\}$, де $n_{\min}$ – мінімальний за замовчуванням обсяг вибірки, та тестову $Y_{test} = \{y^{(t)}_i \mid t = \overline{1, T};\ \ i = \overline{1, N}\}$ матриці.

Крок 3. Запуск інформаційно-ектремального навчання за навчальною матрицею Y з квазіоп-тимізацією параметра рецептивних полів $\delta_s = \delta_{si}$ за процедурою (3) при стартових параметрах $\{\delta^{start}_{s_i}\}$.

Крок 4. $\delta^{start}_{s_i} := \delta^*_s$.

Крок 5. Ініціалізація лічильника векторів-реалізацій тестової матриці Y_{test}: $t := 0$.

Крок 6. Запуск інформаційно-ектремального навчання з послідовною оптимізацією параметра рецептивних полів $\{\delta_{si}\}$ за процедурою (3.4.4) при стартових параметрах $\{\delta^{start}_{s_i}\}$.

Крок 7. $\delta_{s_i}^{start} := \delta_{s_i}^{*}$.

Крок 8. $t := t + 1$.

Крок 9. Якщо $t \le T$ то визначити належність $y^{(t)}$ до одного з класів алфавіту $\{X_m^o\}$ за правилом (3.4.7) , інакше перехід до кроку 11.

Крок 10. Якщо належність вектора-реалізації $y^{(t)}$ не співпадає з апріорною класифікацією, то додати $y^{(t)}$ до навчальної матриці Y та перейти до кроку 6, інакше до кроку 8.

Крок 11. ЗУПИН.

Таким чином, алгоритм навчання класифікатора трафіку в рамках IEI-технології полягає в ітераційній процедурі наближення глобального максимуму інформаційного КФЕ (3.4.2) до його граничного значення шляхом оптимізації параметрів рецептивних полів та геометричних параметрів контейнерів на кожному ярусі ієрархічної структури вирішальних правил. При цьому отримані вирішальні правила за навчальною вибіркою малого обсягу використовуються для сканування тестової вибірки з метою донавчання при невірних класифікаціях.

Розглянемо результати реалізації запропонованого алгоритму навчання інформаційно-екстремального класифікатора для ідентифікації зашифрованого трафіку таких застосувань як SSH (трафік дистанційного керування операційними системами та тунелювання TCP-з'єднань), VoIP (представлений в основному голосовим трафіком Skype) та BitTorrent (зашифрований трафік пірінгової мережі обміну файлами), оскільки вони останнім часом займають значну частину від загального мережевого трафіку і враховуються адміністратором при настройці QoS-механізму.

Навчальні набори даних були сформовані в процесі трасування трафіку утилітою TcpDump [79] з наступним формуванням потоків і обчисленням ознак розпізнавання за допомогою утиліти NetMate [72, 79]. Апріорна класифікація реалізацій навчального трафіку основана на

результатах моніторингу сокетів утилітою CurrPorts (для Windows) [72] та Net Activity Viewer (для Linux) [80].

Ознаками розпізнавання є статистичні характеристики двонаправленого потоку пакетів, де як потік розглядається ряд пакетів, що поділяють однаковий кортеж з п’яти елементів: IP-адреса джерела та отримувача, номер портів джерела і отримувача, номер протоколу. При цьому TCP-потоки обмежені тривалістю до 600 с, а UDP-потоки обмежені максимальною тривалістю між прибуттям пакетів, що становить 64 с. Загальна кількість ознак розпізнавання становить $N = 37$, а саме: кількість пакетів та байтів в прямому/зворотному напрямках потоку; відношення кількості пакетів до кількості байтів корисного навантаження в прямому/ зворотному напрямках; середнє значення, мінімальне значення, перша та третя квартілі, медіана та дисперсія розміру корисного навантаження (в байтах) для вхідних/вихідних пакетів двонаправленого потоку; відношення кількості пакетів малого розміру (до 50 байтів корисного навантаження) до загальної кількості пакетів в прямому/зворотному та в обох напрямах; відношення кількості пакетів великого розміру (більше 1300 байтів корисного навантаження) до загальної кількості пакетів в прямому/зворотному та в обох напрямах; мінімальне, максимальне та середнє значення тривалості часового інтервалу між прибуттям пакетів в прямому/зворотному напрямах; відношення кількості пакетів без корисного навантаження до загальної кількості пакетів в прямому/зворотному та в обох напрямах; кількість прапорців ACK / PSH в потоці прямого/зворотного напряму.

Збір даних моніторингу мережевого трафіку здійснювався на 10-ти комп’ютерах локальної мережі кафедри комп’ютерних наук Сумського державного університету протягом 5 годин робочого дня. Загальний обсяг накопичених наборів даних становить 5,31 Гбайт (22 271 754 реалізацій). Набір даних незбалансований, проте була здійснена процедура видалення однакових реалізацій, що зменшило обсяг даних до 4,11 Гбайт (17 238 589

реалізацій). Для побудови в процесі виконання алгоритму LEARNING-1 ієрархічної структури вирішальних правил (рис.3.8) для чотирьох класів розпізнавання використано випадкові вибірки з наявного набору даних по 100 векторів-реалізацій для кожного класу.

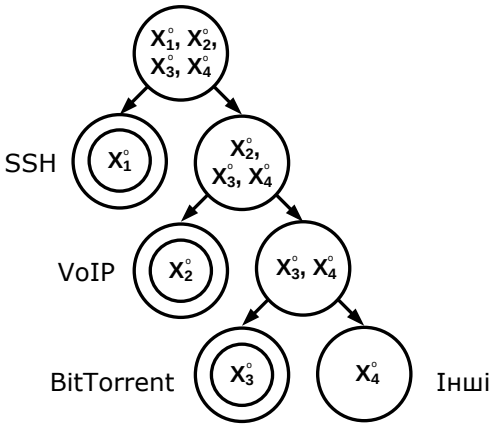
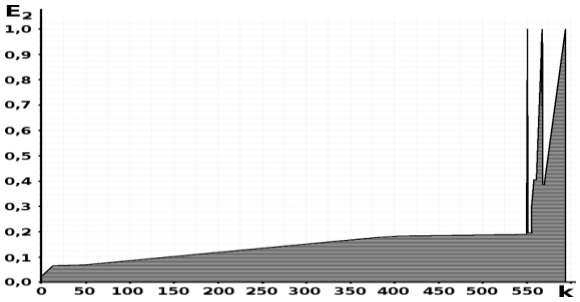


Рисунок 3.8– Ієрархічна структура вирішальних правил для чотирьох класів розпізнавання

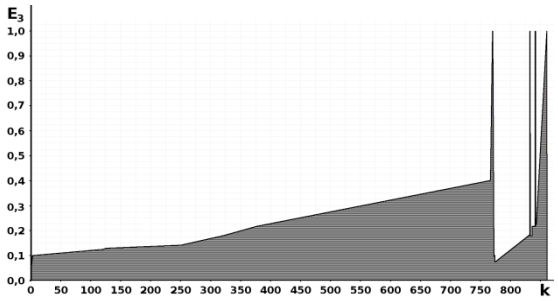
Аналіз рис. 3.8 показує, що на першому ярусі побудовано контейнер для класу X_1^o (трафік SSH), на другому ярусі побудовано контейнер для класу X_2^o (трафік VoIP), а на третьому ярусі побудовано контейнер для класу X_3^o (трафік BitTorrent). Клас X_4^o є сусіднім до класу X_3^o і представлений реалізаціями трафіку, які характеризують веб-серфінг, незашифрований обмін файлами та деякі фонові процеси оновлення системного та користувацького програмного забезпечення, DNS-запити та інше.

З метою підвищення точності класифікатора реалізується його донавчання за алгоритмом LEARNING-2. Графіки зміни максимумів нормованого КФЕ в процесі послідовної оптимізації параметра рецептивних полів на кожному ярусі ієрархічної структури в режимах навчання (до першого максимуму $E_s^o = 1{,}0$) за алгоритмом LEARNING-1 та донавчання (після першого максимуму $E_s^o = 1{,}0$) за алгоритмом LEARNING-2 показано на рис. 3.9. При цьому кожен крок оптимізації

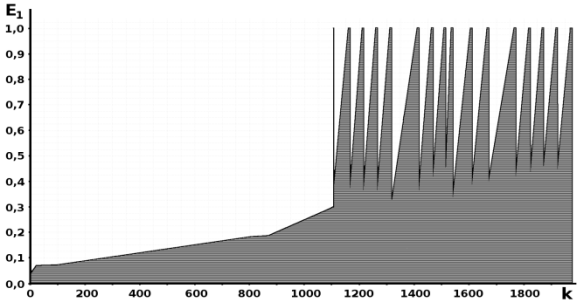
збільшує лічильник кроків k і відповідає одній зміні параметра рецептивного поля для будь-якої ознаки.



а



б



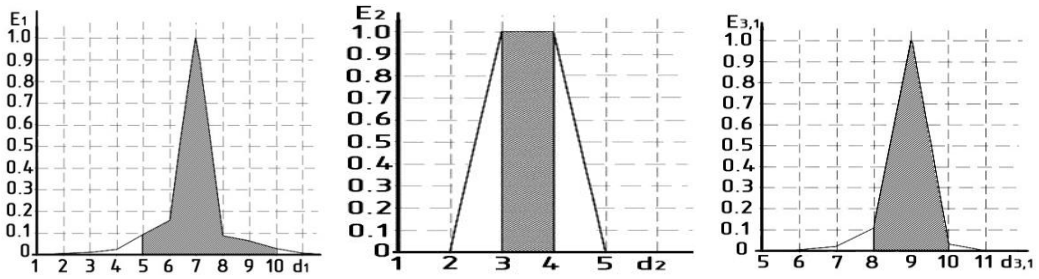
в

Рисунок 3.9– Графіки зміни максимумів КФЕ в процесі оптимізації рецептивних полів: *а* – перший ярус; *б* – другий ярус; *в* – третій ярус

Аналіз рис. 3.9 показує, що в процесі навчання на кожному ярусі було отримано безпомилкові за навчальною матрицею вирішальні правила. Але в процесі донавчання за алгоритмом LEARNING-2 на першому ярусі було виявлено дві помилкові класифікації тестових реалізацій, на другому ярусі – три помилкові класифікації і на третьому ярусі – 15 помилкових класифікацій. Після додавання до навчальної матриці помилково класифікованих реалізацій вдалося побудувати

безпомилковий за навчальною та тестовою матрицями класифікатор.

На рис. 3.10 показано графіки залежності радіусу гіперсферичного контейнера для кожного класу, що розпізнається на s-му ярусі при оптимальних рецептивних полях для ознак розпізнавання.



а *б* *в*

Рисунок 3.10 – Графіки залежності нормованого КФЕ від радіусів контейнерів класів розпізнавання: *а* – клас X_1^o; *б* – клас X_2^o; *в* – клас X_3^o

Аналіз рис. 3.10 показує, що оптимальні значення радіусів контейнерів для класів розпізнавання дорівнюють: для класу X_1^o – $d_1^* = 7$ (тут і далі в кодових одиницях), для класу X_2^o – $d_2^* = 3$, для класу X_3^o – $d_3^* = 9$.

Таким чином, запропонований алгоритм інформаційно-ектремального навчання дозволяє отримати безпомилкові вирішальні правила за незбалансованими навчальними вибірками для розпізнавання зашифрованного трафіку SSH, VoIP та BitTorrent.

3.5 Метод прогнозування рівнів сприйняття якості обслуговування в інфокомунікаційних системах

З розвитком інформаційно-телекомунікаційних систем потреби клієнтів і їх поведінка змінилися. Центр уваги зміщується від підвищення продуктивності мережі до покращення сприйняття якості обслуговування (Quality of Experience, QoE). Забезпечення заданого рівня QoE для послуг і додатків стає фундаментальною задачею при реалізації наскрізного керування ресурсами [51, 55]. На рівень QoE впливають технологічні, соціальні та економічні фактори. При цьому технологічні фактори пов'язані з якістю обробки зростаючих обсягів трафіка, а соціальні характеризують вид контенту (аудіо, відео, інше), область застосування (освіта, мистецтво, медицина, інше), спосіб надання (streaming, broadcast, файл, інше), напрямок потоків (однонаправлені, двонаправлені, мультінаправлені), місце знаходження (в транспорті, вдома, на вулиці, інше) та пристроєм доступу (смартфон, ноутбук, планшет, інше). Економічні фактори пов'язані з бажанням постачальників збільшити свій прибуток, зберегти існуючих та залучити нових клієнтів шляхом розвитку позитивного чи негативного досвіду користувачів відносно тих чи інших послуг через рекламу, програми лояльності та інші методи впливу. В працях [55, 56] показано, що зв'язок QoE з традиційними ключовими метриками продуктивності (Key Performance Indicators) інформаційно-телекомунікаційних систем має нелінійний, нестаціонарний та, часто, неоднозначний характер. Як було показано в праці [51] перспективним підходом для оцінювання QoE є застосування ідей і методів машинного навчання та розпізнавання образів. При цьому врахування контекстних факторів через доступні метадані та результати опитувань клієнтів дозволяє побудувати більш точні моделі прогнозування QoE і реалізувати цінову диференціацію надання послуг. В праці [81] пропонується перехід від угод про рівень послуг (Service Level Agreements, SLA) до угод про

рівень очікуваної якості послуг (Experience Level Agreements, ELA). При цьому перешкодою до широкого впровадження угод ELA є відсутність єдиного погляду на питання формування вхідного математичного опису систем оцінювання QoE для різноманітних інформаційно-телекомунікаційних сервісів та недослідженість проблеми оптимізації пов'язаних з цим витрат і відповідно прибутку. Основна причина такого стану обумовлена невирішеними проблемами науково-методологічного характеру, пов'язаними з незавершеністю формування основ функціонально-вартісного аналізу.

Одним з перспективних шляхів інформаційного синтезу систем керування інфокомунікаційними сервісами є застосування функціонально-вартісного аналізу, в якому оптимізація функціональної складової здійснюється в рамках інформаційно-екстремальної інтелектуальної технології аналізу даних.

Розглянемо в рамках функціонально-вартісного аналізу задача синтезу інформаційно-екстремального класифікатора рівнів QoE, здатного адаптуватися до умов і потреб ринку інформаційно-телекомунікаційних послуг.

Нехай дано алфавіт класів розпізнавання $\{X_m^o \mid m=\overline{1,M}\}$, які описують якість сприйняття сервісу інформаційно-телекомунікаційної системи. Як реалізації кожного класу розпізнавання розглядаються впорядковані вектори ознак, до складу яких входять метрики продуктивності та дані контекстних факторів впливу на QoE. Відомі матриця $\{C_{m,k} \mid m=\overline{1,M}; k=\overline{1,M}\}$, що містить втрати внаслідок помилкового розпізнавання реалізації класу X_m^o як реалізації класу X_k^o та ціна W одного опитування щодо рівня якості сервісу. Відомий структурований вектор параметрів функціонування класифікатора

$$g =< n_m, \Sigma, \delta_i, d_m >, \qquad (3.5.1)$$

де n_m – кількість векторів-реалізацій у вибірці m-го класу для потреб навчання та перенавчання; Σ – словник ознак розпізнавання; δ_i – параметр, що визначає для i-ї ознаки ширину поля контрольних допусків; d_m – кодовий радіус контейнера класу X_m^o, що відновлюється в радіальному базисі бінарного простору ознак $\Omega_Б$.

При цьому задано такі обмеження: $n_m \geq n_{\min}$, де $n_{\min}$ – мінімальний репрезентативний обсяг вибірки для навчання (перенавчання); $d_m < d(x_m \oplus x_c)$, де $d(x_m \oplus x_c)$ – кодова відстань між двійковим еталонним (усередненим) вектором класу X_m^o та двійковим еталонним (усередненим) вектором сусіднього класу X_c^o; $\delta_i \in [0; \delta_{\max}]$, де $\delta_{\max}$ – граничне значення параметра поля контрольних допусків.

Необхідно в процесі навчання системи сформувати навчальну матриця типу “об’єкт-властивість” $\left\{ y_{m,i}^{(j)} \mid m = \overline{1, M};\ j = \overline{1, n_m};\ i = \overline{1, N} \right\}$, де N – потужність повного словника ознак розпізнаванн, та визначити оптимальні значення координат вектора параметрів функціонування (3.5.1), що забезпечують максимальне значення узагальненого критерію функціональної ефективності (КФЕ)

$$J = \frac{\overline{E}}{E_{\max}} \cdot \frac{C_{\min}}{C_{training} + C_{error}}, \qquad (3.5.2)$$

де $\overline{E}$ – усереднене за алфавітом класів $\{X_m^o\}$ значення інформаційного критерію ефективності машинного навчання класифікатора рівнів QoE; $E_{\max}$ – максимальне граничне значення інформаційного критерію; $C_{\min}$ – мінімальне граничне значення витрат оператора/провайдера, пов’язаних з експлуатацією системи оцінювання рівня QoE; $C_{training}$ – значення затрат на експлуатацію системи, що включає, наприклад, витрати на опитування

респондентів і отримання контекстних ознак, вартість системних ресурсів, задіяних при навчанні (перенавчанні) та екзамені; C_{error} – втрати оператора/провайдера інформаційно-телекомунікаційної системи, пов'язані зі штрафами та неоптимальним керуванням IT-інфраструктурою внаслідок помилкового оцінювання рівня QoE.

Необхідно в режимі екзамену, тобто в робочому режимі системи оцінювання якості інформаційно-телекомунікаційних сервісів, прийняти рішення про належність вектора-реалізації до одного з класів алфавіту $\{X_m^o\}$, сформованого на етапі навчання, з метою прогнозування рівня QoE.

Розглянемо алгоритм функціонування здатного навчатися класифікатора.

Кодування навчальної матриці в рамках IEI-технології здійснюється з урахуванням ймовірнісних характеристик як кількісних, так і категоріальних ознак в базовому класі $X_Б^o \in \{X_m^o\}$, за правилами:

$$x_{m,i}^{(j)} = \begin{cases} 1, & якщо \;\; A_{L,i} \le y_{m,i}^{(j)} \le A_{H,i}; \\ 0, & інакше \end{cases}$$

$$x_{m,C*i-r}^{(j)} = \begin{cases} 1, \text{ якщо } y_{m,i}^{(j)} = v_r \;\; та \;\; A_{L,i} \le f_{m,i,r}^{(j)} \le A_{H,i}; \\ 0, \text{ інакше} \end{cases} r = \overline{1,C},$$

де $y_{m,i}^{(j)}$ – числове значення i-ї ознаки в j-й реалізації m-го класу розпізнавання; $A_{L,i}$, $A_{H,i}$ – нижній та верхній пороги системи контрольних допусків на ознаки розпізнавання; C – кількість номінальних значень, що може приймати і - категоріальна ознака; v_r – r-е значення категоріальної ознаки; $f_{m,i,r}^{(j)}$ – відносна частота появи r-го значення в i-й категоріальній ознаці.

Процедура двійкового кодування i-ї ознаки потребує обчислення верхньої $A_{H,i}$ та нижньої $A_{L,i}$ меж поля контрольних допусків, що здійснюється за правилами:

$$A_{B,i} = \bar{y}_{Б,i} + \delta_i;\ A_{H,i} = \bar{y}_{Б,i} - \delta_i,$$

де δ_i – заданий параметр поля контрольних допусків для i-ї ознаки розпізнавання; $\bar{y}_{Б,i}$ – середнє вибіркове значення ознаки – частоти появи категоріальної ознаки в базовому класі $X_Б^o \in \{X_m^o\}$.

Ітераційна процедура оптимізації впорядкованого вектора параметрів поля контрольних допусків $<\delta_1, \delta_2, .. \delta_i, .., \delta_{N-1}, \delta_N>$ на значення ознак та вектора ймовірностей $<p_1, p_2, .. p_i, .., p_{N-1}, p_N>$ включення ознак до робочого словника Σ^* полягає в максимізації узагальненого критерію (3.5.2)

$$\Sigma^* = \arg \max_{\Sigma \in \Omega} \{ \max_{G_\delta} \{ \max_{\{k\}} J_k \}, \qquad (3.5.3)$$

де G_δ – область допустимих значень поля контрольних допусків; {k} – множина кроків оптимізації геометричних параметрів розбиття.

Як критерій ефективності машинного навчання класифікатора розглядається модифікація інформаційної міри Кульбака

$$E_m = \left[1 - 2[p_1 \alpha_m^{(k)} + p_2 \beta_m^{(k)}]\right] * \log_2 \left[\frac{1 - [p_1 \alpha_m^{(k)} + p_2 \beta_m^{(k)}]}{p_1 \alpha_m^{(k)} + p_2 \beta_m^{(k)}} \right] \qquad (3.5.4)$$

де α_m − помилка другого роду при розпізнаванні реалізацій класу X_m^o; β_m − помилка другого роду; p_1, p_2 –безумовні ймовірності появи

реалізацій класу X_m^o та найближчого до нього сусіднього класу X_c^o відповідно.

Робоча (допустима) область визначення функції інформаційного критерію обмежена нерівностями $D_{1,m} = 1 - \alpha_m \geq 0{,}5$, $D_{2,m} = 1 - \beta_m \geq 0{,}5$.

При обчисленні витрат, пов'язаних з використанням системи оцінювання рівнів QoE в даному дослідженні запропоновано обмежитися затратами на формування вхідного математичного опису та втратами внаслідок неточної оцінки рівня QoE:

$$C = C_{training} + C_{error} = W * \sum_{m=1}^{M} n_m + \sum_{m=1}^{M} \sum_{k=1}^{M} C_{m,k} * n_{m,k},$$

де $n_{m,k}$ – кількість реалізацій класу X_m^o, що потрапляють до контейнера класу X_k^o.

При оцінці втрат враховується, що за відсутності машинного навчання рішення приймається на основі контролю виконання умов контрактів на обслуговування (SLA), а після формування вирішальних правил штрафи за порушення умов SLA знімаються тільки при порушенні умов ELA, тобто при відсутності адекватної реакції на зниження QoE.

Оскільки в задачах багатопараметричної оптимізації окрім точності обчислень набуває важливого значення задача підвищення оперативності, то для оптимізації в рамках IEI-технології параметрів функціонування класифікатора було обрано роєвий алгоритм пошуку косяком риб (Fish School Search, FSS), який на відміну від інших алгоритмів еволюційного програмування, основаних на імітації механізмів природи, відрізняється простотою реалізації, інтерпретабельністю та високою швидкістю збіжності.

В алгоритмі FSS кожна риба зберігає одне з рішень задачі. При цьому косяк риб є агрегацією агентів рою, які рухаються приблизно з

однією й тією ж швидкістю і орієнтацією, підтримуючи приблизно однакову відстань між собою. Індивідуальний успіх кожної риби в пошуці рішення характеризується її вагою, що відіграє роль пам'яті. Кожна ітерація пошуку виконує дві групи операторів – оператори годування та оператори плавання.

Оператор годування формалізує успішність дослідження агентами тих чи інших областей "акваріуму" і полягає в обчисленні ваги z-го агента, яка пропорційна нормалізованій різниці значень фітнес-функції на наступній та поточній ітераціях

$$w_z[k+1] = w_z[k] + \frac{J(P_z[k+1]) - J(P_z[k])}{\max(J(P_z[k+1]), J(P_z[k]))},\ z = \overline{1, Z},$$

де $P_z[k+1]$, $P_z[k]$ – позиція z-го агента в багатовимірному просторі рішень на k-й та $(k+1)$-й ітерації агоритму FSS.

Максимально можливе значення ваги агента w_z у алгоритмі FSS обмежується значенням $w_{\max} > 0.$ При цьому під час ініціалізації популяції всім агентам присвоюється вага, що рівна $w_{\max} \cdot 0{,}5.$

В алгоритмі FSS розрізняють три види плавання – індивідуальне, інстинктивно-колективне та колективно-вольове. Ці види плавання здійснюються послідовно один за одним в окремі інтервали часу $(t, \tau], (\tau, \theta], (\theta, t^{'})$, $t < \tau < \theta < t^{'}, t^{'} = t + 1.$

Під час індивідуального плавання агентів відбувається їх переміщення, що має рівноймовірний випадковий характер. При цьому за одну ітерацію алгоритму FSS крок індивідуального плавання виконується фіксовану кількість разів. Компоненти кроку переміщення V_z^{ind} рівномірно розподілені в заданому інтервалі $v_{\max}^{ind}$

$$V_z^{ind} = U(0;1)v_{\max}^{ind},\ z = \overline{1,Z},$$

де $U(0;1)$ – випадкове число із заданого діапазону $(0;1)$.

В процесі інстинктивно-колективного плавання на кожного з агентів чинять вплив всі інші агенти популяції і цей вплив пропорційний індивідуальним успіхам агентів. При цьому позиції агентів обчислюються за формулою

$$P_z^{\theta} = P_z^{\tau} + \frac{\sum_j V_j^{ind}(\tau)(J(P_j^{\tau}) - J(P_j^{t}))}{\sum_j J(P_j^{\tau}) - J(P_j^{t})},\ z = \overline{1,Z}. \qquad (3.5.5)$$

Колективно-вольове плавання полягає у зміщенні всіх агентів у напрямку поточного центру тяжіння популяції за умови збільшення сумарної ваги косяка риб в результаті індивідуального та інстинктивно-колективного плавання. Якщо сумарна вага зменшилась, то зміщення відбувається в протилежному напрямку. Колективно-вольове плавання виконується за правилами

$$\mathrm{P_z^{'}} = \mathrm{P_z^{\theta}} \pm \mathrm{v^{vol}}(\mathrm{P_z^{\theta}} - \mathrm{P_c^{\theta}}),\ \mathrm{z} = \overline{1,\mathrm{Z}}, \qquad (3.5.6)$$

де P_c^{θ} – координати центра тяжіння косяка риб, що визначаються за формулою

$$P_c^{\theta} = \frac{\sum_z w_z^{\theta} P_z^{\theta}}{\sum_z w_z^{\theta}}.$$

У формулі (3.5.6) знак плюс використовується за умови

$$\sum_z w_z^{\theta} > \sum_z w_z^{\theta-1},$$

а знак мінус – в протилежному випадку. При цьому розмір кроку переміщення агентів v^{vol} є випадкою величиною

$$v^{vol} = v^{vol}_{\max} U(0;1),$$

де $v^{vol}_{\max}$ –максимально допустима довжини кроку переміщення при колективно-вольовому плаванні.

Оптимізація затрат на машинне навчання здійснюється в процесі максимізації узагальненого критерію (3.5.2). При цьому умовою зупину є зменшення критерію (3.5.2 на значення, що перевищує заданий поріг Δ.

Розглянемо основні кроки алгоритму оптимізації затрат на навчання.

1. Обчислення мінімальних витрат на обслуговування системи оцінювання QoE.
2. Формування навчальних вибірок мінімального обсягу з максимальним словником ознак для заданого алфавіту класів.
3. Ініціалізація лічильник кроків збільшення затрат на машинне навчання: $k := 0$;
4. Додаткові опитування респондентів;
5. Запуск алгоритму інформаційно-екстремального навчання (3.5.3).
6. $k := k + 1$
7. Якщо $J[k] > J\max[k-1]$, то $J\max[k] := J[k]$, інакше – $J\max[k] := J\max[k-1]$.
8. Якщо $J_{\max}[k] - J[k] > \Delta$, де Δ – заданий поріг, то перехід до кроку 9, інакше – до кроку 4.
9. ЗУПИН

У режимі екзамену рішення про належність вектора-реалізації $x^{(j)}$ до одного з класів алфавіту $\{X^o_m\}$ приймається шляхом обчислення геометричної функції належності

$$\mu_m = 1 - \frac{d(x_m^* \oplus x^{(j)})}{d_m},$$

де $d(x_m^* \oplus x^{(j)})$ – кодова відстань від центра k -го контейнера класу X_m^o до вектора $x^{(j)}$.

Таким чином, алгоритм машинного навчання в рамках функціонально-вартісного аналізу полягає у реалізації ітераційної процедури максимізації узагальненого критерію ефективності системи (2) з метою формування оптимальних в інформаційному та варісному сенсі вхідного математичного опису та відповідних вирішальних правил.

За даними моніторингу процесу обслуговування мобільних клієнтів сервіс-провайдера мультимедійного контенту було сформовано вибірки спостережень для трьох класів сприйняття якості обслуговування QoE. Клас X_1^o характеризував неприйнятний (незадовільний) рівень якості, а класи X_2^o і X_3^o характеризували прийнятний та відмінний рівні якості обслуговування відповідно. Обсяг вибірки кожного класу становив $n_{\max} = 400$, а потужність повного словника ознак – $N = 25$. До словника ознак включено ключові та інтегральні показники продуктивності компонентів IT-інфраструктури, задіяних у наданні сервісу, та контекстні ознаки, що характеризують тип, популярність та ціну мультимедійного контенту, місце розташування користувача, тип пристрою перегляду, тип мережевого з’єдання, час доби та день тижня.

На рис. 3.11 показано графік оптимізації затрат на машинне навчання класифікатора рівнів QoE за узагальненим критерієм (3.5.2).

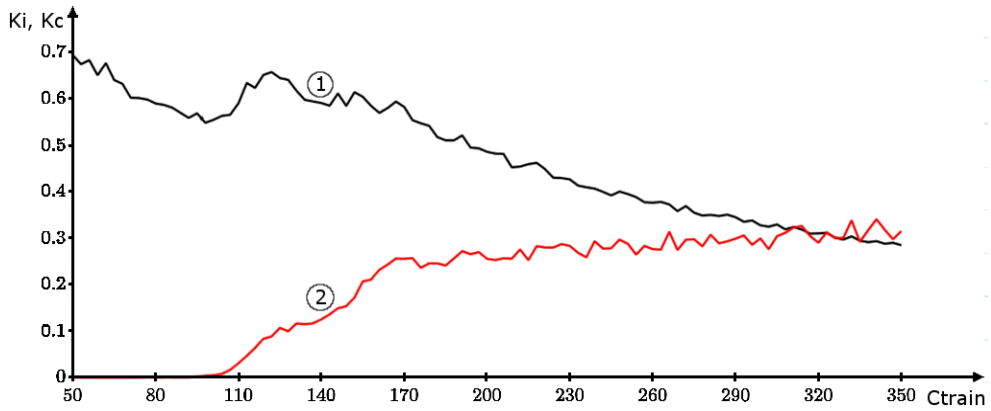


Рисунок 3.11 – Залежність інформаційної та вартісної складової узагальненого критерію (3.5.2) від затрат на машинне навчання

Аналіз рис. 3.11 показує, що підвищення затрат на машинне навчання дозволяє дещо підвищити достовірність вирішальних правил і відповідно зменшити втрати внаслідок не виправдання очікувань клієнтів щодо якості сервісу чи простою невикористаних ресурсів. При цьому починаючи з деякого рівня затрат, вони стають не виправданими, оскільки істотно не підвищують достовірність вирішальних правил. На рис. 3.12 показано графік зміни узагальненого критерію (3.5.2) від затрат на машинне навчання.

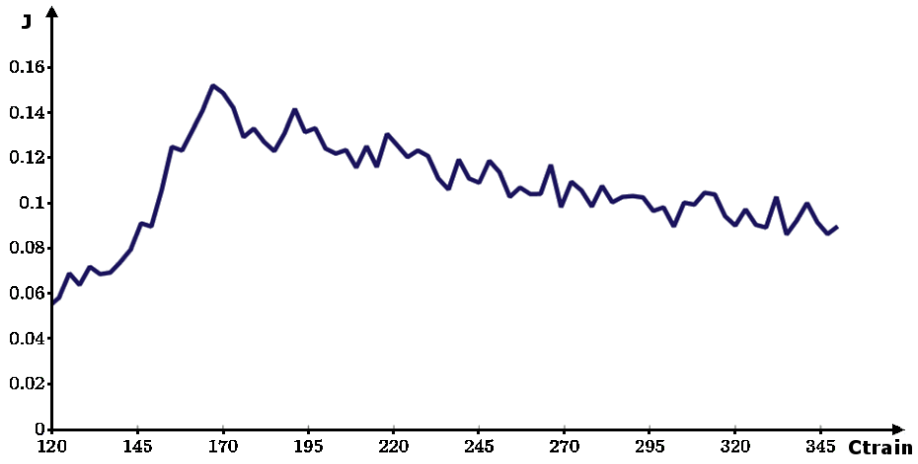


Рисунок 3.12 – Графік залежності узагальненого критерію (3.5.2) від затрат на навчання класифікатора

Графік приведений на рис. 3.12 побудований на повному наборі даних з метою ілюстрації закономірності, яка полягає в поступовому наростанні значення критерію та наступному його спаданні. В практичних (не лабораторних) умовах процес розширення навчальної матриці припиниться раніше. Момент зупину визначається заданим порогом Δ, що визначає максимальне значення зменшення критерію (3.5.2) порівняно з глобальним максимумом.

Аналіз рис. 3.12 показує, що починаючи з відмітки $C_{train} = 165$ значення критерію (3.5.2) має тенденцію до спадання і в одному з локальних мінімумів буде прийнято рішення про зупин процедури навчання і перехід в робочий режим оцінювання рівня QoE.

На рис. 3.13 показано графік зміни максимумів усередненого за алфавітом класів інформаційного критерію K_I в процесі роєвої оптимізації косяком риб багатовимірного вектора параметрів, що включає в себе ймовірності входження ознак розпізнавання до робочого словника та значення параметру поля контрольних допусків для кожної ознаки розпізнавання, при оптимальному обсязі навчальної матриці.

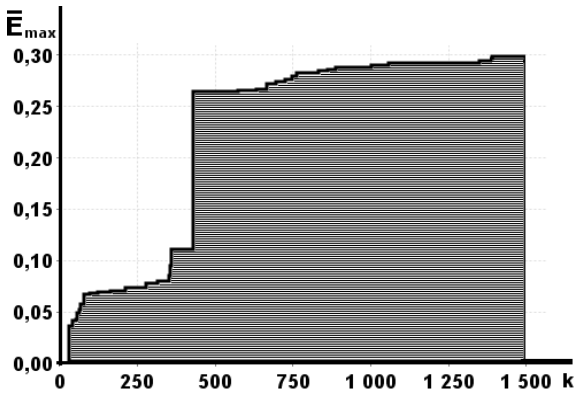
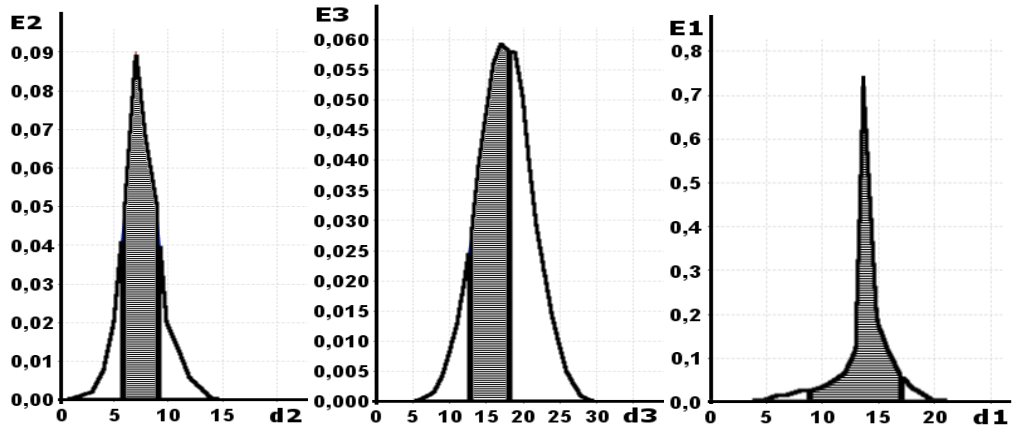


Рисунок 3.13 – Залежність КФЕ від кількості переміщень агентів роєвого пошуку

Аналіз рис. 3.13 показує, що в результаті 1500 ітерацій алгоритму

пошуку було знайдено оптимальні в інформаційному сенсі словник ознак та параметри полів контрольних допусків на значення ознак розпізнавання. При цьому максимальне значення усередненого нормованого інформаційного критерію не досягає граничного значення.

На рис. 3.14 показано результати оптимізації геометричних параметрів контейнерів класів розпізнавання.



а *б* *в*

Рисунок 3.14 – Графіки залежності інформаційного критерію від радіусу контейнера класу: *а* – клас X_1^o; *б* – клас X_2^o; *в* – клас X_3^o

Аналіз рис. 3.14 показує, що оптимальні значення радіусів гіперсферичних контейнерів класів X_1^o, X_2^o і X_3^o відповідно дорівнюють: $d_1^* = 14$, $d_2^* = 7$ і $d_3^* = 17$ (в кодових одиницях). При цьому значення інформаційного критерію для класів X_1^o, X_2^o і X_3^o становлять $E_1^* = 0{,}9$, $E_2^* = 0{,}06$ і $E_3^* = 0{,}075$, що відповідає таким відповідним значенням ймовірностей правильного розпізнавання $P_{true,1} = 0{,}99$, $P_{true,2} = 0{,}8$, $P_{true,2} = 0{,}75$.

Таким чином, отримані в процесі оптимізації параметрів вектора функціонування (3.5.1) вирішальні правила для прогнозування рівня QoE є оптимальними в інформаційному та вартісному сенсі.

ВИСНОВКИ

У науково-дослідній роботі розв’язано важливу науково-технічну прикладну задачу аналізу і синтезу здатної навчатися системи керування інфокомунікаційним сервісом. Головні наукові та практичні результати роботи полягають у такому:

1. Основним способом енергозбереження в сучасних розподілених обчислювальних середовищах є використання технології віртуалізації та алгоритмів перерозподілу навантаження між фізичними серверами для мінімізації їх простою. При цьому існує ризик порушення умов SLA, тому для реалізація цього способу необхідно додатково використовувати методи прогнозування функціонального стану фізичних серверів та віртуальних машин.

2. Виконаний в роботі аналіз праць попередників показав, що в сучасних інфокомунікаційних системах все частіше впроваджуються засоби інформаційної безпеки, основані на ідеях і методах машинного навчання та розпізнавання образів. Архітектура системи виявлення вторгнень набуває ієрархічної структури, на всіх рівнях якої знаходяться здатні до взаємодії інтелектуальні автономні агенти. При цьому керування безпекою інформаційно-комунікаційної системи полягає у своєчасній реакції на мережеві атаки та забезпеченні економічно доцільного рівня захищеності об’єктів системи відповідно до їх інформаційної цінності.

3. З метою оптимізації використання ресурсів інфраструктури в рамках реалізації окремих сервісів, планування потреб в ресурсах, уникнення відмов та збоїв, підтримки заданого рівня задоволеності користувачів, оцінки ймовірності успіху внесення різноманітних змін в середовище тощо запропоновано узагальнений критерій ефективності системи керування інфокомунікаційним сервісом. При цьому основною складовою узагальненого критерію ефективності системи керування є

інформаційний критерій якості її машинного навчання, а вартісна складова узагальненого критерію є допоміжною, оскільки вона служить для контролю за доцільністю матеріальних витрат на подальше навчання (перенавчання) системи.

4. Оскільки синтез прогностичних вирішальних правил системи керування інформаційно-телекомунікаційним середовищем та безпосереднє прийняття рішень передбачають пошук екстремуму функції критерію багатьох параметрів, то запропоновано здійснювати пошук глобального оптимуму оптимізаційної задачі даного типу з використанням ідей і методів ройового інтелекту.

5. В рамках IEIT-технології синтезовано класифікатор мережевих трафіків з підвищеною достовірністю та оперативністю. Оперативність навчання класифікатора підвищена за рахунок використання отриманих вирішальних правил в процесі навчання за вибірками малого розміру. При цьому побудовані безпомилкові за навчальною матрицею вирішальні правила у рамках геометричного підходу забезпечують прийняття рішень в робочому режимі в реальному темпі часу.

За результатами фізичного моделювання за даними моніторингу трафіків SSH, VoIP та BitTorrent було доведено високу ефективність розроблених алгоритмів і отримано безпомилкові за навчальними матрицями вирішальні правила.

6. В рамках функціонально-вартісного аналізу здійснено синтез інформаційно-екстремального класифікатора рівнів QoE, що враховує як інтегральні показники продуктивності компонентів IT-інфраструктури, так і контекстні ознаки. При цьому не вдалося отримати безпомилкові за навчальною матрицею вирішальні правила, що пов’язано із суб’єктивною природою даних опитування та недостатньою інформативністю словника ознак. Тому дослідження буде продовжено в напрямку розширення словника інформативними ознаками.

ПЕРЕЛІК ПОСИЛАНЬ

1. Довбиш А.С. Основи проектування інтелектуальних систем: Навчальний посібник / А.С. Довбиш // Суми: Видавництво СумДУ, 2009.– 171 с.
2. Довбыш А.С. Информационно-экстремальный алгоритм оптимизации параметров гиперэллипсоидных контейнеров классов распознавания / А.С. Довбиш, Н.Н. Будник, В.В Москаленко // Проблемы управления и информатики: Междунар. науч.-техн. журнал. − 2012.− № 5. − С. 111−119.
3. Довбиш А.С. GRID-сервисный центр телекоммуникационной системы распознавания изображений медицинских объектов / А.С. Довбиш, С.С. Мартиненко // Межд. Крымская конференция “СВЧ техника и телекоммуникационные технологии”. Материалы конференции в 2 т..- Севастополь: Вебер.-2010.-Т.1.-С. 490-491 (Scopus).
4. Dovbysh A.S. Information-extreme algorithm for recognizing current distribution maps in magnetocardiography / A.S. Dovbysh, S.S. Martynenko, A.S. Kovalenko, N.N. Budnyk // Journal of Automation and Information Sciences .– 2011.– V. 43.– № 2.–P. 63-70 (SCOPUS).
5. Chaykovskyy I. Development of pattern recognition method for diagnosis of myocardial ischemia and noncoronarogenic myocardial diseases based on current density distribution maps / I. Chaykovskyy, M. Budnyk, A. Dovbysh, A. Kovalenko et al // Advances in Biomagnetism, Int. Federation Med&Biol Engineering (IFMBE) Proceedings Series. – 2010. – Vol. 28. – Springer Verlag: Heidelberg. – pp. 424-427 (SCOPUS).
6. Dovbysh A.S. Information-extreme cluster-analisis for self-learning GRID-centers / A.S. Dovbysh, S.A.S.M. Julgam // Microwave and Telecommunication Technology (CriMiCo), 22nd International Crimean

Conference, 8-12 September. – New York: Wiley-IEEE Press. − 2012. – P. 413 – 414 (SCOPUS).

7. Dovbysh A.S. Information-extreme algorithm for estimation of telecommunication GRID-system nodes / A.S. Dovbysh, V.V. Moskalenko //Microwave and Telecommunication Technology (CriMiCo), 2012 22nd International Crimean Conference. – New York : Wiley-IEEE Press. − P. 415 – 416 (SCOPUS).
8. Dovbysh A.S. Information-Extreme Algorithm for Optimizing Parameters of Hyperellipsoidal Containers of Recognition Classes / A.S. Dovbysh, N.N. Budnyk, V.V. Moskalenko // Journal of automation and information sciences.− New York: Begell House Inc. − v44.i10.−2012.−P. 35-44 (SCOPUS).
9. Довбиш А.С. Иерархический информационно-экстремальный алгоритм обучения системы диагностирования с избыточной обучающей матрицей /А.С. Довбыш, А.А. Стадник, С.А. Голубничий // Проблемы управления и информатики, 2015. – №2. – С. 70-80.
10. Довбиш А.С. Інформаційно-екстремальний алгоритм навчання системи підтримки прийняття рішень з гіперціліндроїдним класифікатором / А.С. Довбиш, Г.А. Стадник // Радіоелектронні і комп’ютерні системи. – 2015. – С. 36-44.
11. Довбиш А.С. Оптимізація словника ознак розпізнавання для інформаційно-екстремального гіпереліпсоїдного класифікатора / А.С. Довбиш, В.В. Москаленко // Вісник Національного технічного університету “Харківський політехнічний інститут”. – Харків: НТУУ “ХПІ”. – 2012. − № 30. – С. 65 – 77.
12. Москаленко В.В. Інформаційно-екстремальний унімодальний класифікатор з паралельно послідовною оптимізацією контрольних допусків на ознаки розпізнавання / І.В.Шелехов, О.В.Соболєв, В.В.

Москаленко// Автоматизовані системи управління та прилади автоматики. – Харків: ХНУРЕ. – 2011. − №156. − С. 42 – 47.

13. Москаленко В.В. Інформаційно-екстремальне навчання системи підтримки прийняття рішень з адаптивною кластеризацією даних / В.В. Москаленко // Вісник Сумського державного університету. Суми : СумДУ. – 2012. №3. С. 92 – 106.
14. Москаленко В.В. Прогнозування моменту перенавчання інформаційно-екстремального унімодального класифікатора / В.В.Москаленко // Біоніка інтелекту. – Харків : ХНУРЕ. – 2012. – №2(79). – С. 47 – 52.
15. Москаленко В.В. Оптимізація словника ознак розпізнавання для інформаційно екстремального унімодального класифікатора / І.В.Кузьмін, В.В. Москаленко // Вісник Вінницького політехнічного інституту. – Вінниця : ВНТУ. – 2012. – №5. – С. 91 – 97.
16. Москаленко В.В. Ієрархічний інформаційно-екстремальний класифікатор / В.В. Москаленко, Джулгам С.А.С.М // Радіоелектронні і комп’ютерні системи // Харків: НАКУ “ХАІ”. – 2012. − №3(55). − С. 86 – 93.
17. Москаленко В.В. Информационно-экстремальный алгоритм оптимизации параметров гиперэллипсоидных контейнеров классов распознавания / В.В. Москаленко, А.С. Довбиш, Н.Н.Будник //Проблемы управления и информатики: Междунар. науч.-техн. журнал. − 2012.− № 5. − С. 111−119.
18. Москаленко В.В. Інформаційно-екстремальна кластеризація даних Автоматизовані системи управління та прилади автоматики / В.В Москаленко // – Харків: ХНУРЕ. – 2012. − №160. − С. 75 – 84.
19. Москаленко В.В. Інтелектуальна автоматизована система керування з оптимізацією часових параметрів аналізу вхідних даних / В. В. Москаленко, А.С.Рижова // Вісник Сумського державного університету. – Суми : СумДУ. – 2013. – №3. – С. 7 – 14.

20. Москаленко В.В. Інформаційно-екстремальний метод ідентифікації типу мережевого трафіка / В.В. Москаленко, А.С. Рижова // Радіоелектронні та комп'ютерні системи. – Харків: НАКУ“ХАІ”. – 2014. − №4(68). − С. 138 – 144.
21. Москаленко В.В. Інформаційно-екстремальний класифікатор з комбінованими контейнерами для планування задач в розподіленому гетерогенному обчислювальному середовищі / В.В. Москаленко, А.С. Рижова // Радіоелектронні та комп'ютерні системи. – Харків: НАКУ “ХАІ”. – 2015. − №2(72). − С. 77 – 87.
22. Москаленко В.В. Информационно-экстремальное машинное обучение по несбалансированным данным большого раз мера / В.В. Москаленко // Біоніка інтелекту. – Харків :ХНУРЕ. – 2015. – №1 (84). – С. 34 – 38.
23. Москаленко В.В. Інтелектуальна система прогнозування зниження продуктивності віртуальних машин в середовищі хмарних обчислень / В.В. Москаленко, С.В. Пимоненко // Journal of engineering sciences. – Суми : СумДУ. – 2015. – №1. – F.1-F.7.
24. Москаленко В.В. Інтелектуальна діагностична система для радіонуклідного статичного обстеження / В.В. Москаленко, А. С. Рижова // Journal of engineering sciences. – Суми : СумДУ. – 2015. – №2. – H1.-H8.
25. Москаленко В.В. Интеллектуальная система поддержки принятия решений с оптимизацией пространственно-временных параметров функционирования / В.В. Москаленко // Автоматизовані системи управління та прилади автоматики. – Харків: ХНУРЕ. – 2014. − №169. − С. 42 – 50.
26. Москаленко В.В. Система прогнозування перевантажень та відмов в хмарних сервісах розподіленої обробки великих даних / В.В. Москаленко, С.В. Пимоненко // Радіоелектронні та комп'ютерні системи. – Харків: НАКУ “ХАІ”. – 2015. − №2(72). − С. 77 – 87.

27. Москаленко В.В. Информационно-экстремальный метод классификации наблюдений с категориальными даннями (Information-extreme method of classification of observation with categorical attributes / А.С. Довбиш, В.В. Москаленко, А.С. Рижова // Кібернетика і системний аналіз. – Київ : Київ: Інститут кібернетики ім. В.М. Глушкова НАНУ. – 2016. – №2 (52). – С.4-13. Kibernetika i sistemny analiz. – Kyiv, Ukraine.
28. Москаленко В.В. Система поддержки принятия решений для интеллектуального управления нестационарным технологическим процессом / А.С. Довбиш, А.С. Рижова, В.В. Москаленко // Проблемы управления и информатики. – Киев: Институт кибернетики им. В.М. Глушкова НАНУ. – 2016. – №2. – С.5-13.
29. Москаленко В.В. Інформаційно-екстремальний алгоритм оцінки стану вузлів телекомунікаційної GRID- системи / В.В. Москаленко, А.С. Довбиш // Тези доповідей 22-ї Міжнародної Кримської конференції «Мікрохвильова техніка і телекомунікаційні технології». – Севастополь: Вебер. – Т.1. − 2012. – С. 415 – 416.
30. Москаленко В.В. Категорійна модель навчання інформаційно-екстремального гіпереліпсоїдного класифікатора / В.В. Москаленко // Матеріали VIII Міжнародної конференції «Стратегія якості у промисловості і освіті». – Дніпропетровськ-Варна:ТУ-Варна. – Т.3. – 2012. – С. 459 – 460.
31. Москаленко В.В. Ієрархічний інформаційно-екстремальний алгоритм навчання системи підтримки прийняття рішень / В.В. Москаленко // Тези доповідей V Міжнародної науково-технічної конференції «Комп'ютерні системи та мережеві технології». – Київ : НАУ. – 2012. – С. 100.
32. Москаленко В. В. Категорійна модель інформаційно-екстремальної кластеризації даних / В. В. Москаленко // Тези доповідей 4-ї Міжнародної науково-практичної конференції «Інтелектуальні

системи в промисловості і освіті» – Суми: СумДУ. – 2013. − С. 109 – 112.

33. Москаленко В.В. Інформаційно-екстремальне навчання за вибірками з різнотипними ознаками розпізнавання / В.В. Москаленко, А.С. Рижова // Матеріали XI міжнародної наукової конференції «Інтелектуальні системи прийняття рішень і проблеми обчислюваного інтелекту». – Херсон: ХНТУ. – 2015. – С. 304-306.
34. Москаленко В.В. Інтелектуальна система керування розподілом ресурсів телекомунікаційної мережі /В.В. Москаленко, С.В. Пимоненко // Матеріали III міжнародної науково-практичної конференції «Обчислювальний інтелект». – Черкаси : ЧДТУ. – 2015. – С. 249.
35. Довбыш А.С. Информационно-экстремальный метод классификации наблюдений с категориальными признаками / А.С. Довбиш, В.В. Москаленко, А.С. Рыжова // Кибернетика и системный анализ. – 2016. – Том 52. – №2. – С. 56-63.
36. Dovbysh A. S. Information-Extreme Method for Classification of Observations with Categorical Attributes / A. S. Dovbysh, V. V. Moskalenko, A. C. Rizhova // Cibernetica and Systems Analysis. – 2016. – V. 52. – №2. – P. 45-52 (SCOPUS).
37. Dovbysh, A. S. Designing algorithms for optimization of parameters of functioning of intelligent system for radionuclide myocardial diagnostics [Text] / A. S. Dovbysh, A. S. Moskalenko, V. V. Moskalenko, I. V. Shelehov // Informa-tion and controlling system. – Eastern-European Journal of Enterprise Technologies. – 2016. – № 3/9(81). – P. 11–18 (SCOPUS).
38. Dovbysh A. Information synthesis of adapti9ve system for visual diagnostics of emotional and mental state of a person / A. Dovbysh, I Shelehov, D. Prylepa, I. Golub // Eastern-European Journal of Enterprise Technologies. – 2016. – Vol. 4/9 (82). – P. 11-17 (SCOPUS).

39. Довбиш А.С. Інформаційно-екстремальне навчання системи якості навчального контенту випускової кафедри / А.С. Довбиш, Є. С. Кулік, З. В. Козлов, А. С. Осадчий // Радіоелектронні і комп'ютерні системи: наук.-техн. журнал. − 2016.− №3(71).− С. 70-77.
40. Васильєв А.В. Інформаційно-аналітична система адаптації навчального контенту випускової кафедри до вимог ринку праці/ А. В. Васильєв, А. С. Довбиш, Є. С. Кулік, З. В. Козлов // Бионика интеллекта. – 2016.–№2(87). – С. 23-30.
41. Васильєв А.В. Інформаційний синтез інформаційно-аналітичної системи оцінки якості навчального контенту / А. В. Васильєв, А.С. Довбиш, В.О. Любчак, А. С. Осадчий // Радіоелектронні і комп'ютерні системи: наук.-техн. журнал. − 2016.− №4(72).− С. 36-43.
42. Dovbysh A.S. Improving Information and Software Support for Data Mining System of Drones / A.S. Dovbysh, J.V. Symonovskiy, O.V. Korobchenko // Advanced Information Systems and Technologies: proceedings of the IV international scientific conference, May 25-27, 2016 / Ред.кол.: С.І. Проценко, В.В. Шендрик, С.М. Ващенко. – Sumy: Sumy State University, 2016. – С. 104–105.
43. Довбиш А.С. Досвід розробки інтелектуальних систем керування в Сумському державному університеті / А. С Довбиш // Матеріали XXIII міжнародної конференції з автоматичного управління (Автоматика-2016), м. Суми, 22-23 вересня 2016 року. – Суми: Сумський державний університет, 2016. – С. 10–11.
44. Довбиш А.С. Інформаційно-екстремальний алгоритм машинного навчання системи розпізнавання транспортних засобів / А.С. Довбиш, Ю.В. Симоновський, О.В. Коробченко, М.А. Летюга.// Вісник НТУ «ХПІ», 2016. – № 45(1217). – С. 17–22.

45. Москаленко В.В. Вступ до інформаційного аналізу і синтезу інфокомунікаційних систем : навч. посіб. / В. В. Москаленко, А.С. Довбиш. – Суми : Сумський державний університет, 2016. – 226 с.
46. Грушин Д. А. Энергоеффективные вычисления для группы кластеров / Д. А. Грушин, Н. Н. Козюрин // Труды института системного программирования РАН, 2012. –Т. 23. –С. 433 – 445.
47. Helmy T. Independent Job Scheduling by Fuzzy C-Means Clustering and Ant Optimization Algorithm in a Computation GRID / T. Helmy, Z. Rasheed / International Journal of Computer Science. – 2010. – V. 37, I.3. – Massachusetts, USA: MIT-Press. – P. 531-536.
48. Nanduri R. Job Aware Scheduling Algorithm for MapReduce Framework / R. Nanduri, N. Maheshwari, R. Raja, V. Varma // 3rd IEEE International Conference on Cloud Computing Technology and Science. – Athens, Greece : IEEE Press, 2011. – P. 724–729.
49. Карпенко А. П. Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой / А. П. Карпенко. – Москва : МГТУ им. Н. Э. Баумана, 2014. – 446 с.
50. SLA Management Handbook / TM Forum Publication. – Morristown, NJ, USA : TeleManagement Forum, 2001. – 141 p.
51. Varela M. From Service Level Agreements (SLA) to Experience Level Agreements (ELA): The challenges of selling QoE to the user [Text] / M. Varela, P. Zwickl, M. Xie, H. Schulzrinne [et al.] // Communication Workshop (ICCW), 2015 IEEE International Conference on 8–12 June 2015. – London, UK : IEEE Press, 2015. – P. 1741–1746.
52. Мерков А. Б. Распознавание образов. Введение в методы статистического обучения / А. Б. Мерков. – Москва : URSS, 2011. – 254 с.
53. Chen L. A Multitarget Heuristic Algorithm f or Virtual Machine Placement / L. Chen, J. Zhang, L. Cai, R. Li, T. He, T. Meng //

International Journal of Distributed Sensor Networks. – New York, NY, USA : Hindawi Publishing Corporation, 2015. – Vol. 2015 – 14 p.

54. Кальпеева Ж. Б. Модели и методы организации вычислительных процессов в распределенной облачной среде : дис. … доктора философии (PhD) : 6D070400 / Ж. Б. Кальпеева. – Алматы, Республика Казахстан, 2014. – 136 с.

55. Alreshoodi M. Survey on QoE/QoS correlation models for multimedia services [Text] / M. Alreshoodi, J. Woods // International journal of distributed and parallel systems. – Bristol, PA, Taylor & Francis Inc.– 2013. – Vol. 4, №. 3. – P. 53–72.

56. Skorin-Kapov L. A Multi-Dimensional View of QoE: the ARCU Model / L. Skorin-Kapov, M. Varela // Proceedings of the 35th International Convention MIPRO. – 2012. – Optija, Crotia : IEEE Press. – P. 662 – 666.

57. 57.Mateo P. A Context-aware Model for the Analysis of User Interaction and QoE in Mobile Environments / P. Mateo, D. S. Ruiz, G. M. Perez // International Journal of Human-Computer Interaction. – 2014. – Vol. 30, I. 12. – Norwood, N.J, USA : Ablex Pub. – P. 946-964.

58. Теленик С. Ф. Зведення метрик оцінювання рівня обслуговування користувачів на основі експертних оцінок [Текст] / С. Ф. Теленик, О. І. Ролік, О. М. Моргаль, О. С. Квітко // Вісник Вінницького політехнічного інституту. – 2011. – № 1. – С. 112–123.

59. Балькин Г. Ф. Системный анализ в инфокоммуникациях : учебное пособие / Г. Ф. Балькин, Ю. Г. Балькин, Л. А. Крапивянская // Государственный университет телекоммуникаций. – Киев : ГУТ МОН Украины, 2014. – 97 с.

60. Pedro M. Santiago del Rio. Internet Traffic Classification for High-Performance and Off-The-Shelf Systems / Pedro Maria Santiago del Rio // Ph.D. Thesis. – 2013. – Madrid, Spain : Technical University of Madrid. – 217 p.

61. Zhao W. Internet Quality of Service: an Overview / W. Zhao, D. Olshefski, H. Schulzrinne // Columbia University Research Report CUCS-003-00. – Manhattan, New York, USA : IBM && Columbia, 2003. – 11 p.
62. Bujlow T. Volunteer-Based System for classification of traffic in computer networks / T. Bujlow, K. Balachandran, M.T. Riaz, J.M. Pedersen // In Proceedings of 19th Telecommunications Forum TELFOR 2011'.– 2011.– Sydney : IEEE Press.– P. 210-213.
63. Vladusic D. Using the black-box approach with machine learning methods in order to improve job scheduling in GRID environments / D. Vladusic // WSEAS Transactions on Systems and Control. – . – 2009. – i.6,v.4. – P. 259-270.
64. Garcıa J.L.B. Improved Self-management of DataCenter Systems Applying Machine Learning // J.L.B. Garcıa / Ph.D.Thesis. – Barcelona, Catalunya (Spain) : Polytechnic University of Catalonia. – 2013. – 155 p.
65. Zander S. Automated Traffic Classification and Application Identification using Machine Learning / S. Zander, T. Nguyen, G. Armitage // Proceedings of the IEEE Conference on Local Computer Networks 30th Anniversary. – 2005. – Sydney : IEEE Press. – P. 250-257.
66. Chakravarty S. Traffic Analysis Attacks and Defenses in Low Latency Anonymous Communication / S. Chakravarty // PhD thesis Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Graduate School of Arts and Sciences. – Manhattan, New York, USA : Columbia University, 2014. – 138 p.
67. Кузьмин И. В. Оценка эффективности и оптимизация автоматизированных систем контроля и управления.– Москва : Сов. радио, 1971. – 296 с.

68. Cerioli A. Robust classification with categorical variables / A. Cerioli, M. Riani, A. C. Atkinson // Proceedings in Computational Statistics. − Heidelberg : Physica-Verlag HD, 2006. − P. 507–519.
69. Зайченко Ю. П. Основи проектування інтелектуальних систем: Навчальний посібник для вузов / Ю. П. Зайченко. – Київ : Слово, 2004 . – 352 с.
70. Довбиш А. С. Основи теорії розпізнавання образів : навч. посіб.: у 2-х ч. Ч.1 / А. С. Довбиш, І. В. Шелехов. – Суми: СумДУ, 2015. – 109 с.
71. Довбиш А. С. Інтелектуальні інформаційні технології в електронному навчанні: монографія / А. С. Довбиш, А. В. Васильєв, В. О. Любчак. – Суми: СумДУ, 2013. – 177 с.
72. Raman Singh. Issue related to sampling techniques for network traffic dataset / Raman Sigh, Harish Kumar, R.K. Singla //International Journal of Mobile Network Communications & Telematics. – 2013. – Sydney, Australla : WSP. – Vol.3., No.4. – P. 75-85.
73. Singh R. Issue related to sampling techniques for network traffic dataset / R. Sigh, H. Kumar, R. K. Singla // International Journal of Mobile Network Communications & Telematics. – Sydney, Australla : WSP, 2013– Vol. 3, № 4. – P. 75–85.
74. Yap B.W. An Application of Oversampling, Undersampling, Bagging and Boosting in Handling Imbalanced Datasets / B.W. Yap, K. A. Rani, H. A. A. Rahman, S. Fong, Z. Khairudin. N.N. Abdullah // Proceedings of the First International Conference on Advanced Data and Information Engineering. Lecture Notes in Electrical Engineering, 2014. – V. 285. – Singapore : Springer Science. – P. 13-22.
75. Загоруйко Н. Г. Прикладные методы анализа данных и знаний / Н. Г. Загоруйко. − Новосибирск : ИМ СО РАН, 1999. – 270 с.
76. Шелехов І. В. Інформаційно- екстремальна оптимізація словника ознак розпізнавання / І. В. Шелехов, М. М. Бірюкова // Вісник

Сумського державного університету. Серія технічні науки – Суми, 2012. – № 3. – С. 46–54.

77. Довбиш А.С. Оптимізація словника ознак розпізнавання системи керування, що навчається / А.С. Довбиш, І.В. Шелехов, О.В. Коробченко // Міжвідомчий науково-технічний збірник «Адаптивні системи автоматичного управління». – 2015. – №2(27). – 44-50.
78. Russell Eberhart Swarm Intelligence / Russell Eberhart, Yuhui Shi, James Kennedy. – Morgan Kaufmann, 2001. – 512 p.
79. Iacovazzi A. Network Communication Privacy: Traffic Masking against Traffic Analysis /Alfonso Iacovazzi // Ph.D. Thesis. – 2013. – Rome, Italy: Sapienza University of Rome. 119 p.
80. Markowsky G. Who's Knocking at Your Cybercastle's Gate? / G. Markowsky, L. Markowsky // In Proceedings of International Conference on Security and Management. – 2012. – Las Vegas Nevada, USA : DCSREA Press. – P. 206-212.
81. Bouten N. QoE Optimization Through In-Network Quality Adaptation for HTTP Adaptive Streaming / N. Bouten, J. Famaey, S. Latre, R. Huysegems, B. D. Vleeschauwer, W. Van Leekwijck, F. D. Turck // Proceedings of the 8th International Conference on Network and Service Management. – 2012. – Las Vegas, USA : IEEE Press, Research Publishing Services. – P. 336-342.