МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет імені Петра Могили

Факультет комп'ютерних наук

Кафедра інженерії програмного забезпечення

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри інженерії програмного забезпечення, канд. техн. наук, доцент,

_____Є. О. Давиденко

«____»_____2024 р.

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

АВТОМАТИЗОВАНЕ ДІАГНОСТУВАННЯ ПРИЧИН ЗБОЮ МЕРЕЖІ

Спеціальність «Інженерія програмного забезпечення»

121 – КРМ.1 − 608м.22250825

Здобувач _____А. А. Андрєєв

підпис

«___» _____ 2024 р.

Керівник канд. пед. наук, доцент кафедри інженерії програмного забезпечення

_____К. О. Кірей

підпис

«___» _____ 2024 р.

Миколаїв – 2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра інженерії програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри інженерії програмного

забезпечення, канд.техн.наук, доцент,

_____Є. О. Давиденко

«___» _____ 2023 р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи магістра

Видано студенту групи 608м факультету комп'ютерних наук

Андрєєв Андрій Андрійович .
(прізвище, ім'я, по батькові студента)

1. Тема кваліфікаційної роботи

«Автоматизоване діагностування причин збою мережі» .

Затверджена наказом по ЧНУ від «10» листопада 2023 р. № 234

2. Строк представлення кваліфікаційної роботи «__» _____ 2024 р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні

Вхідні дані до роботи − функціональні та нефункціональні вимоги до інформаційної системи моніторингу збоїв у мережі. Результат – функціонуюча інформаційна система для автоматизованого діагностування збою мережі .

4. Перелік питань, що підлягають розробці:

- аналіз систем діагностики збоїв у мережі;
- розробка проєктних рішень для діагностування збоїв мережі;
- проєктування та моделювання системи;
- програмна реалізація автоматизованої системи діагностики збоїв в мережі;

5. Перелік графічних матеріалів:

Презентація.

Керівник роботи канд. пед. наук, доцент Кірей Катерина Олександрівна
(посада, прізвище, ім'я, по батькові)

_____
(підпис)

Завдання прийнято до виконання

Андрєєв Андрій Андрійович
(прізвище, ім'я, по батькові студента)

_____
(підпис)

Дата видачі завдання «___» _____ 2023 р.

КАЛЕНДАРНИЙ ПЛАН
виконання кваліфікаційної роботи

Тема: «Автоматизоване діагностування причин збою мережі» .

№	Найменування роботи	Початок	Закінчення	Примітки
1.	Розробка та затвердження завдання на виконання КРМ	10.11.2023	13.11.2023	Виконано
2.	Огляд літератури за темою роботи	14.11.2023	21.11.2023	Виконано
3.	Складання календарного плану КРМ	22.11.2023	25.11.2023	Виконано
4.	Аналіз предметної області	27.11.2023	07.12.2023	Виконано
5.	Розробка проєктних рішень	08.12.2023	18.12.2023	Виконано
6.	Моделювання та конструювання ПЗ	18.12.2023	28.12.2023	Виконано
7.	Кодування, тестування та апробація розробленого ПЗ, аналіз результатів тестування, розробка керівництва користувача	29.12.2023	20.01.2024	Виконано
8.	Відгук керівника КРМ	28.01.2024	31.01.2024	Виконано
9.	Оформлення КРМ та презентації	03.02.2024	06.02.2024	Виконано
10.	Попередній захист	07.02.2024	08.02.2024	Виконано
11.	Рецензування	09.02.2024	15.02.2024	Виконано
12.	Завершення оформлення КРМ та презентації	15.02.2024	20.02.2024	Виконано
13.	Захист кваліфікаційної роботи	26.02.2024	27.02.2024	Виконано

Розробив студент Андрєєв Андрій Андрійович _____
(прізвище, ім'я, по батькові студента) (*підпис*)

«___» _____ 2023 р.

Керівник роботи канд. пед. наук, доцент Кірей К. О. _____
(посада, прізвище, ім'я, по батькові) (*підпис*)

«___» _____ 2023 р.

АНОТАЦІЯ

до кваліфікаційної роботи магістра

«Автоматизоване діагностування причин збою мережі»

Студент 608м гр.: Андрєєв Андрій Андрійович

Керівник: канд. пед. наук, доцент Кірей К. О.

В умовах проведення цифрової трансформації важливо захистити носії інформації та канали зв’язку від несанкціонованого доступу на всіх етапах комунікаційного процесу. Тому актуальним є питання посилення контролю за станом телекомунікаційної системи та виявлення причин збоїв у роботі комп’ютерної мережі підприємства при використанні каналів електронних комунікацій, мережевих ресурсів та інформаційних продуктів.

Об’єкт дослідження – діяльність по забезпеченню стабільності телекомунікаційних систем в розрізі швидкого виявлення причин збоїв в мережі. Предмет дослідження – забезпечення контролю стану телекомунікаційних систем шляхом використання системи діагностики причин збоїв. Метою роботи є розробка інформаційної системи діагностики стану вузлів в телекомунікаційних мережах для дослідження ймовірності виникнення збоїв мережі

В першому розділі здійснено аналіз систем діагностики збою мережі. У другому розділі розроблено проєктні рішення для діагностики збоїв мережі. В третьому розділі спроєктовано автоматизовану систему діагностики збою мережі. В четвертому розділі виконана програмна реалізація інформаційної системи. Розроблено керівництво користувача.

В результаті виконання кваліфікаційної роботи магістра реалізована інформаційна система автоматизованого діагностування причин збою мережі.

КРМ викладена на 63 сторінки, вона містить 4 розділи, 20 рисунків, 6 таблиць, 58 джерел в переліку посилань.

Ключові слова: комп’ютерні мережі, захист даних, Simulink, нечіткі множини.

ABSTRACT

of the Master’s Thesis

«Automated diagnosis of network failure causes»

Student of group 608: Andrieiev Andrii Andriiovych

Supervisor: Dr.Sc., Docent Kirei K. O.

In the conditions of digital transformation, it is important to protect media information and communication channels from unauthorized access at all stages communication process. Therefore, the issue of strengthening control over the state of the telecommunications system and identifying the causes of malfunctions computers' network of the enterprise when using electronic channels communications, network resources and information products.

The object of the research is activity to ensure stability telecommunication systems to quickly identify the causes of failures in network The subject of the study is the provision of condition control telecommunication systems by using the diagnostic system causes of failures. The purpose of the work is to develop an information diagnostic system state of nodes in telecommunication networks for probability research occurrence of network failures

In the first section, an analysis of network failure diagnostics systems was carried out. In the second section, project solutions for diagnosing network failures are developed.

In the third section, an automated failure diagnosis system is designed network. In the fourth section, the software implementation of the information system is performed. A user manual has been developed.

As a result of the completion of the master`s qualification work, the master`s degree was realized information system for automated diagnosis of network failure causes.

The work is 63 pages long and includes 4 sections, 20 illustrations, 6 tables, and 58 sources.

Keywords: *computer networks, data protection, Simulink, fuzzy sets*

ЗМІСТ

ВСТУП

Актуальність теми. В сучасній цифровій епохі, де інформація переважно зберігається та обмінюється через електронні канали зв'язку, захист телекомунікаційних мереж і інформаційних ресурсів від зовнішнього втручання та несанкціонованого доступу стає критично важливою проблемою. Ця актуальність постає через потенційні загрози, такі як крадіжка конфіденційної інформації, розголошення комерційних таємниць, а також можливість спотворення або знищення важливих даних. Тому вирішальною стає необхідність зміцнення контролю за доступом і запобігання несанкціонованому втручанню в роботу телекомунікаційних систем та інформаційних ресурсів. У цьому контексті особливо важливим стає виявлення причин можливих збоїв у роботі комп'ютерних мереж та інфраструктури підприємств при використанні електронних комунікаційних каналів та інформаційних продуктів.

Дослідження, що стосуються забезпечення інформаційного забезпечення систем управління та формування механізмів кібербезпеки, привертають увагу значної кількості науковців, як українських (наприклад, Є. Балашова, В. Глушков, В. Гончаров, А. Іваненко, І. Кононенко, Н. Міхеєва, Г. Поспєлов), так і зарубіжних (таких як К. Парк, Т. Пенг, Е. Чен, А. Саммерс). Націленість наукових досліджень на специфіку електронних комунікацій в умовах сучасного господарства виражена у працях таких зарубіжних вчених, як К. Парк, Т. Пенг, Е. Чен, А. Саммерс. Українськими дослідниками, такими як В. Хорошко, В. Чередніченко, Л. Перевалов, С. Кваш, Я. Невойт та інші, проведено значні дослідження з кібербезпеки та заходів захисту від несанкціонованого доступу.

Незважаючи на зацікавленість фахівців у питаннях безпеки в інформаційному секторі, існують значні теоретичні, методологічні і практичні проблеми, які ще не вирішені. Більшість досліджень зосереджуються на аналізі інформації та електронних комунікаціях або на

загальних аспектах безпеки. Проте недостатньо уваги приділяється забезпеченню безпечного доступу до інформаційних ресурсів та реагуванню на загрози несанкціонованого доступу до телекомунікаційних мереж. Ця актуальність проблеми в сфері кібербезпеки підкреслює практичне значення для збереження інформації та необхідність подальшого розвитку в цьому напрямі.

Об’єкт дослідження – діяльність по забезпеченню стабільності телекомунікаційних систем в розрізі виявлення причин збоїв в мережі.

Предмет дослідження – забезпечення контролю стану телекомунікаційних систем шляхом використання системи діагностики причин збоїв.

Метою роботи є розробка інформаційної системи діагностики стану вузлів в телекомунікаційних мережах для дослідження ймовірності виникнення збоїв мережі.

Виходячи з мети, можемо сформулювати наступні **завдання**:

1) здійснити аналіз предметної сфери та аналогічних інформаційних систем.

2) обрати технології для розроблення автоматизованої системи діагностики збоїв в мережі;

3) спроєктувати та реалізувати автоматизовану систему діагностики збоїв в мережі;

4) провести тестування розробленого програмного забезпечення;

5) скласти відповідну документацію.

Методологічна база та підходи дослідження ґрунтуються на використанні різноманітних методів, як загальних, так і спеціалізованих, включаючи аналітичний, аналітико-синтетичний, історичний, компаративний та інші методи наукових досліджень. Застосування історичного методу дозволило розкрити сутність інформаційної безпеки та проаналізувати розвиток заходів для виявлення причин збоїв у мережі упродовж часу.

Наукові концепції, розроблені як вітчизняними, так і зарубіжними вченими, а також нормативно-правові акти України у сфері комп'ютеризації та захисту інформації, становлять теоретико-методологічну основу дослідження. Для аналізу загроз використовувалися аналітичний та аналітико-синтетичний методи, що дозволили систематизувати та оцінити потенційні небезпеки. Компаративний метод застосовано при визначенні рівня ефективності при використанні тих чи інших моделей діагностики.

Дослідження базується на наукових розробках у галузі кібербезпеки, публікаціях як вітчизняних, так і зарубіжних дослідників, а також на досягненнях у сфері системного аналізу, автоматизації управління та захисту обмеженої інформації. Велике значення приділяється також концепціям кібербезпеки, теоріям прийняття рішень в умовах невизначеності й іншим аспектам.

1 АНАЛІЗ СИСТЕМ ДІАГНОСТИКИ ЗБОЇВ В МЕРЕЖІ

1.1 Обґрунтування актуальності проєктування автоматизованої системи діагностики збоїв в мережі

Забезпечення стабільності функціонування комп'ютерної мережі в підприємстві є важливою умовою для ефективності його операцій. Таким чином, будь-які відмови, незалежно від їх походження, порушують цю стабільність і потребують негайної ідентифікації причин та усунення проблем. Отже, швидка реакція та виявлення кореневих причин відмови є ключовими аспектами для запропонованої системи.

Існують два типи систем виявлення: системи виявлення аномалій і системи виявлення ознак. Однак, головною проблемою систем виявлення функцій є їх призначення для виявлення конкретних типів атак, які зазвичай є найнебезпечнішими на момент створення системи. Коли з'являються нові атаки або параметри атак змінюються, системам виявлення потрібно знову адаптуватися.

Системи виявлення аномалій часто базуються на складних моделях нормального інтернет-трафіку, що припускають статистичну однорідність трафіку. Однак не завжди враховується контекст, в якому застосовуються ці припущення, а також умови їх використання. Це може призвести до необхідності перенавчання алгоритмів у випадку навіть незначних змін у структурі трафіку чи надання послуг.

Одним із можливих вирішень даної проблеми полягає у застосуванні комплексного підходу до створення системи протидії атакам. Цей метод включає в себе низку заходів, таких як системний моніторинг, збереження історії транзакцій, створення спеціального репозиторію для проведення інтелектуального аналізу дій зловмисників, а також розробку стратегій протидії.

Інформаційна система буде містити наступні елементи:

1) агенти відстеження;
2) заходи попередньої обробки та зберігання;
3) сховище для зберігання інформації про операції;
4) сховище аналітичних компонентів;
5) заходи проти нападів [2; с. 572].

Для розробки інформаційної системи необхідно визначити математичне забезпечення для кожного компонента:

1) Моніторинг трафіку включає в себе збір пакетів для подальшого аналізу щодо обсягу, характеристик і активності користувачів. Для цього необхідно розробити алгоритми, які визначатимуть оптимальну кількість та частоту захоплення пакетів в залежності від навантаження каналу та інших параметрів. Якщо захоплення пакетів відбувається занадто часто, це може призвести до сповільнення трафіку. У той же час, якщо пакети захоплюються з певними сталими інтервалами, це може створити "сліпі зони", де не буде зібрана інформація.

2) Попереднє опрацювання перехоплених пакетів передбачає оцінку найбільш небезпечних загроз та зберігання інформації у сховищах. У зв'язку з тим, що на цьому етапі потрібна швидка оцінка при мінімальному використанні ресурсів, рекомендується використання простих та адаптивних порогів, або, у випадку потреби, послідовний метод CUSUM.

3) Аналіз даних виконується під час завантаження в пам'ять, з метою виявлення атак та оцінки загрози. Після збереження даних у репозиторії можна провести всебічну оцінку та визначити можливі ризики. Для цього рекомендується скористатися вже відомими методами багатоканального CUSUM і ковзного середнього.

4) Аналіз фонових даних застосовується для виявлення спроб сканування, погіршення якості атак та імпульсних атак. Цей вид аналізу проводиться регулярно або за певним графіком. Оскільки ці види атак

представляють меншу загрозу, виникає можливість проведення більш детального аналізу. Для цього використовуються методи Data Mining, інтелектуальні системи правил, нейронні мережі та інші аналітичні техніки.

5) Прийняття рішення щодо виявлення атаки відбувається у випадку перевищення встановлених порогових значень на одному з попередніх етапів, що може свідчити про можливу загрозу нападу. У такій ситуації налаштовується експертна система для оцінки рівня загрози та прийняття відповідного рішення щодо реагування на атаку.

6) Вибір моделі, оцінка ризику, верифікація і пошук стратегії стають актуальними відразу після виявлення нападу, оскільки необхідно визначити ефективний метод протидії. В залежності від типу та характеристик конкретної атаки, такий протидій може значно варіюватися, що означає наявність «стратегії» протидії. При цьому стратегія може змінюватися залежно від ефективності заходів управління доступом до облікових записів користувачів. Варіанти стратегій мають ґрунтуватися на результаті аналізу взаємодії між нападником та агентами захисту. Вивчення аналітичних моделей допомагає оцінити ефективність контрзаходів та можливі наслідки. Основним фактором, який визначає ігровий процес, є сама конфліктна взаємодія між нападником та системою захисту. Головний впливовий чинник, на який впливають гравці, полягає в навантаженні на систему. Це може включати загальне навантаження або навантаження окремих критичних компонентів системи, таких як процесор, оперативна пам'ять та мережеві канали [2; стор. 573].

Для розробки стратегії протидії необхідно перш за все оцінити параметри конфліктної динамічної моделі. Цей процес включає наступні етапи:

1) Визначення типу атаки.
2) Оцінка кількості зловмисників та їх ресурсів.
3) Оцінка рівня загрози.

4) Визначення можливих запобіжних засобів та способів боротьби з негативними наслідками.

Після використання стратегії захисту система має оцінити її ефективність, шляхом вимірювання загрози та порівняння її з прогнозом.

1.2 Аналіз існуючих засобів діагностики мережі

Методи нечітких множин особливо важливі у випадку відсутності точної математичної моделі функціонування системи. Використання теорії нечітких множин дозволяє враховувати неточні та суб'єктивні експертні знання з конкретної теми для прийняття рішень, не вимагаючи їх формалізації у вигляді традиційних математичних моделей.

Використання теорії нечітких множин дозволяє вирішувати проблеми узгодження конфліктних критеріїв прийняття рішень та розробляти логічні системні регулятори. Нечіткі множини забезпечують можливість використання лінгвістичного опису складних процесів, встановлення нечітких зв'язків між поняттями, прогнозування поведінки системи, створення ряду альтернативних дій та формалізацію нечітких правил прийняття рішень.

Методи теорії нечітких множин застосовуються для розробки інтерфейсів у системах людина-машина. Використання цих методів сприяє управлінню знаннями, підтримці процесів прийняття рішень, апроксимації складних функцій, ідентифікації структури та параметрів систем, розпізнаванню образів та оптимізації процесів. Нечітка логіка застосовується у побутовій електроніці, діагностиці та різноманітних експертних системах. Системи підтримки прийняття рішень на основі нечіткої логіки широко використовуються у військовій, медичній та економічній сферах. Глобальні політичні рішення та кризові ситуації також моделюються з використанням нечіткої логіки. [2, с. 115].

Теорія нечітких множин [4], яку вперше висунув американський математик Лотфі Заде в 1965 році, спрямована на подолання складнощів у розумінні неоднозначних концепцій, аналізу та моделювання систем, в які втручається людина.

Метод, що базується на концепції нечітких множин, фактично є альтернативою стандартним кількісним методам аналізу систем. Його відмінності можна звести до трьох ключових особливостей:

а) використання нечітких величин, так званих "лінгвістичних" змінних, як заміна або додаток до числових змінних;

б) опис зв’язків між змінними за допомогою нечітких висловлювань;

в) використання нечітких алгоритмів для опису складних зв'язків.

Такий підхід є ефективним способом опису поведінки складних систем, які не піддаються точному математичному аналізу з-за певної невизначеності.

Теорія нечітких множин є математичним інструментом для перетворення абстрактних концепцій у числову форму. Нечіткі множини виражають розмиті характеристики або властивості об'єктів, такі як високий, середній, малий тощо. Ця теорія надає методику роботи з нечіткою інформацією, яка дозволяє моделювати людські розуміння та сприйняття. Нечітка логіка включає концепцію часткової істинності, де значення істинності можуть коливатися від 0,0 (повністю помилкові) до 1,0 (повністю правдиві). Тоді нечіткий набір A визначається як множина, функція належності μ_A має значення в діапазоні [0,0, 1,0]. Значення $\mu_A(x) = 0{,}0$ і $\mu_A(x) = 1{,}0$ стоять, відповідно, для нульової та повної належності x до A, тоді як усі значення $\mu_A(x)$ від 0,0 до 1,0 вказують на часткову належність x до множини A. Математично нечіткий набір A представлений функцією належності, визначеною на певному домені X , що називається областю дискурсу:

$$\mu_A : X \to [\,0{,}1\,]$$

де A визначається як нечітка мітка або лінгвістична змінна, яка характеризує змінну x. Як відтворення булевої логіки, μA (x) вказує на ступінь належності x до нечіткого набору A. Нечіткі множини можуть використовуватися для визначення значень нечітких змінних.

Для глибшого осмислення цього інструментарію, давайте звернемося до огляду функцій та структури інструментів нечіткої логіки.

Правила нечіткої логіки дозволяють:

1) застосовувати наявний досвід управління об'єктами;

2) використовувати гнучкі правила, коли точне моделювання системи за допомогою традиційних методів неможливе.

Покращення якості управління досягається за допомогою:

1) автоматичного регулювання системи управління;

2) прогнозування змін у вихідному впливі (функція попередження), заснованого на подіях, які не можуть бути передбачені за допомогою традиційних методів управління.

Кількість програм, які базуються на даних методах управління, постійно зростає як для неперервних, так і для пакетних процесів, а також для автоматизованих систем. Завдяки використанню нечіткої логіки у цій сфері, вона стала методом програмування, що має опис та формулювання. Це дозволяє систематизувати емпіричні знання та використовувати їх для керування процесами у разі ускладнень з використанням традиційних методів управління. Теорія нечіткої логіки дозволяє описати набори методів управління, які легко застосовувати до реальних систем, і враховує досвід операторів і технологів для динамічного керування процесом [20]. Це робить можливим опис окремих етапів виробничого процесу на основі нечіткої логіки, таких як ініціалізація та налаштування параметрів.

Ось декілька важливих характеристик нечіткої логіки:

- гнучка та проста у реалізації техніка машинного навчання;
- допомагає наслідувати логіку людської думки;

- логіка може мати бінарні значення;
- ефективний метод для невизначених або приблизних міркувань;
- нечітка логіка розглядає виводи як процес поширення обмежень;
- нечітка логіка дозволяє створювати нелінійні функції довільної складності;
- для побудови нечіткої логіки необхідне повне керівництво експертів.

Робочий механізм нечіткої системи виходить з конкретного процесу виведення, де змінні, що беруть участь, моделюються за допомогою нечітких наборів.

Використання нечітких наборів дозволяє нечіткій системі відтворювати неоднозначності, що характеризують проблеми реального світу, шляхом формулювання правил типу «ЯКЩО ... ТО ...» на природній мові. Колекція таких правил, відома як база знань, використовується для опису взаємозв'язку між вхідними та вихідними значеннями, а також для визначення відповідного процесу виведення. Нечітка система – це система, що ґрунтується на правилах, які апроксимують невідоме відображення введення/виведення до форми, прийнятної для обробки, використовуючи набір нечітких правил, які оперують зрозумілими для людини твердженнями.

У вирішенні інженерних завдань часто використовується механізм нечіткого виводу Мамдані, який можна програмно реалізувати у середовищі Matlab. Метод Мамдані, який був запропонований Ібрагімом Мамдані в 1975 році, є одним з перших методів, побудованих на основі теорії нечітких множин. Цей метод є нечіткою системою виведення (НСВ), яка використовує теорію нечітких множин для відображення вхідних значень (функцій у разі нечіткої класифікації) на виходи (класи у разі нечіткої класифікації).

Метод Мамдані включає такі етапи:

- формування бази правил;
- фазифікація;

– агрегування підумов;
– активізація підвисновків;
– акумулювання висновків;
– дефазифікація.

Кожен з цих етапів виконується послідовно, причому кожен наступний етап отримує на вхід значення, які були отримані на попередньому етапі. Для визначення вихідних значень метод Мамдані використовує певну базу правил.

Основну структуру нечіткої системи, що була описана Мамдані та Ассіліаном [52], зображено на рис. 1.1. У цій системі чіткі дані подаються на вхід, а система виробляє чіткі дані на виході шляхом виведення з бази нечітких правил, яка є базою знань системи. На початку цієї системи для перетворення чітких даних в нечіткі змінні використовується фазифікатор, а на виході системи – дефазифікатор для перетворення нечітких множин в чіткі значення.

Рисунок 1.1 – Загальна схема системи нечіткого виведення [52]

Механізм нечіткого виведення працює з набором правил, що містяться в базі правил, відповідно до наближеної теорії міркувань. Його завдання - перетворити нечіткі множини у вхідному просторі на відповідні нечіткі множини у вихідному просторі. Таким чином, нечітка система надає обчислювальну схему, яка описує, як правила мають бути оцінені та об'єднані для розрахунку чіткого вихідного значення (вектора) для будь-якого вхідного чіткого значення. Отже, нечітку систему можна розглядати просто як параметричну функцію, яка встановлює відповідність між умовними векторами та реальними векторами.

У системах, що базуються на нечітких правилах, для визначення можуть використовуватися різні моделі, зокрема у системі типу Мамдані. За цієї моделі, кожне правило породжує нечітку множину для лінгвістичної змінної. Висновок правила відбувається за допомогою модус поненс, адаптованого до нечітких множин. Зокрема, з врахуванням вхідного значення, висновок правила полягає в зменшенні нечіткої множини наслідків на величину активації правила, яка визначається ступенем належності вхідного значення до передумов. Визначення оператора "ТО", відомого також як оператор імплікації, визначає вихідну нечітку множину для кожного правила. З використанням нечітких множин у передумові правил, кілька нечітких правил можуть бути активовані одночасно.

Розглянемо основні інструменти нечіткої логіки, які застосовуються в сучасних умовах.

1) Нечіткі нейронні мережі. Нечіткі мережі використовують механізм нечіткої логіки для висновків, але параметри функції належності налаштовуються за допомогою алгоритмів навчання нейронної мережі. Тому для визначення цих параметрів застосовується метод зворотного поширення помилок. Модуль нечіткого управління зазвичай представлений у вигляді багатошарової мережі, що містить чотири рівні: вхідний шар розмиття, рівень агрегації значень активації умови, рівень агрегації нечітких правил і вихідний рівень. На сьогоднішній день найбільш поширеною є архітектура нечіткої нейронної мережі типу AFIS і TSK. Дослідження показали, що такі мережі є універсальними наближеннями. Алгоритми швидкого навчання та можливість інтерпретації накопичених знань роблять нечіткі нейронні мережі одними з найбільш перспективних і ефективних інструментів м'яких обчислень на сьогоднішній день.

2) Адаптивні нечіткі системи. Недолік класичних нечітких систем полягає у складності формування правил та функцій належності. Для цього потрібно мати експертне знання у відповідній галузі, що не завжди можливо

передати. Цю проблему вирішують адаптивні нечіткі системи, де параметри системи вибираються під час навчання на основі експериментальних даних. Алгоритми навчання адаптивних нечітких систем зазвичай складні та трудомісткі, складаючись з двох етапів: генерації правил та коригування функцій належності. Одна з проблем полягає в задачах вичерпного типу, а інша - у оптимізації в неперервних просторах. Виникає протиріччя: функції належності потрібні для генерації нечітких правил, а правила - для нечіткого висновку. Крім того, важливо переконатися, що під час автоматичного створення нечітких правил вони будуть повними та послідовними. Більшість методів навчання нечітких систем використовують генетичні алгоритми (Genetic Fuzzy Systems). Група іспанських дослідників під керівництвом Ф. Еррера внесла значний внесок у розвиток теорії та практики нечітких систем з еволюційною адаптацією.

3) Нечіткі запити. Один зі сприятливих напрямків у сучасних системах обробки інформації – це використання нечітких запитів до баз даних. Цей інструмент дозволяє користувачам формулювати запити природною мовою, наприклад: «Показати список доступних квартир для оренди, які знаходяться в центрі міста та є бюджетними», що є неможливим за допомогою стандартних механізмів запитів. Для цього було розроблено нечітку реляційну алгебру та спеціальні розширення SQL для роботи з нечіткими запитами.

4) Нечіткі асоціативні правила (Fuzzy Association Rules). Це інструмент призначений для отримання та обробки шаблонів із баз даних, які сформульовані у вигляді лінгвістичних виразів. Він включає спеціальні концепції, такі як нечітка транзакція, обробка та валідність правила нечіткої асоціації.

5) Нечіткі когнітивні карти. Використовуються для моделювання причинно-наслідкових зв'язків між поняттями конкретної області, нечіткі когнітивні карти є нечітким спрямованим графом, вузли якого представлені

нечіткими множинами. Спрямовані зв'язки між вершинами графа відображають не лише причинно-наслідкові зв'язки між поняттями, але і визначають ступінь впливу, який одне поняття має на інше. Активне використання нечітких когнітивних карт для моделювання систем обумовлене їхньою здатністю візуально представити аналізовану систему та легко інтерпретувати причинно-наслідкові зв'язки між поняттями. Головні виклики, пов'язані з процесом формування когнітивної карти, полягають у тому, що цей процес не завжди може бути чітко формалізований. Крім того, важливо переконатися, що побудована когнітивна карта відповідає реальній системі. Для розв'язання цих проблем були розроблені алгоритми автоматичної побудови когнітивних карт на основі аналізу наборів даних [23].

6) Нечітке групування. Методи нечіткого групування відрізняються від явних методів, таких як нейронні мережі Кохонена, тим, що дозволяють об'єктам належати одночасно до кількох кластерів з різною ступенем належності. У багатьох випадках нечітке групування вважається більш «природнім», особливо для об'єктів, які знаходяться на межі кластерів. Серед найпоширеніших методів нечіткого групування можна виділити алгоритм самоорганізації нечітких с-середніх, а також його узагальнення у вигляді алгоритму Густафсона-Кесселя. Проте це лише два з численних підходів. Інші популярні методи включають в себе нечіткі дерева рішень, нечіткі мережі Петрі, нечітку асоціативну пам'ять, нечіткі карти самоорганізації та гібридні методи.

Переваги нечіткої логічної системи:

1. Проста і зрозуміла структура нечітких логічних систем.

2. Широке використання нечіткої логіки в комерції та практичних сферах.

3. Можливість контролювати процеси з різними стохастичними параметрами за допомогою правил нечіткої логіки.

4. Можливість використання єдиного прийнятного міркування замість точних міркувань.

5. Допомога у вирішенні питань, пов'язаних з невизначеністю в точних розрахунках.

6. Надійність, оскільки не вимагає введення точних даних.

7. Можливість програмного впровадження в ситуаціях, коли датчик зворотного зв'язку відмовляє.

Недоліки нечітких систем:

1. Нечітка логіка не завжди точна, що може призвести до прийняття результатів на основі припущень та обмеження їхнього широкого прийняття.

2. Відсутність можливості машинного навчання розпізнавання шаблонів нейронними мережами в нечітких системах.

3. Потреба у широкому тестуванні обладнання для перевірки нечіткої системи, що базується на онтологічних знаннях.

4. Складність встановлення точних результатів на основі нечітких правил та функцій.

5. Плутанина нечіткої логіки з теорією ймовірностей.

Загалом, використання нечіткої логіки для отримання наближених оцінок непараметризованих вхідних даних є універсальним. Отже, доцільно розглядати окремі моделі, такі як лінійна модель, яка використовується для апроксимації відображень, у системах визначення причин збою мережі. Ця модель може бути представлена як у математичному, так і у нейронному вигляді, дозволяючи ефективно вирішувати складні завдання.

Пакет ST Neural Networks надає можливість створювати лінійну мережу та навчати її за допомогою стандартного алгоритму лінійної оптимізації на основі псевдообернених матриць (SVD).

Лінійна мережа використовується для оцінки якості побудованих нейронних мереж. Деякі завдання можуть бути успішно вирішені не тільки за допомогою нейронної мережі, але і простим лінійним методом. Якщо

кількість навчальних даних обмежена, то можливо, немає потреби використовувати складніші моделі.

Для побудови моделі інтелектуального класифікатора рекомендується використовувати кожен з описаних типів нейронних мереж. На основі отриманих результатів можна визначити найкращу модель інтелектуального класифікатора.

1.3 Постановка задачі по розробці автоматизованої системи діагностики збоїв в мережі

Показники надійності визначаються шляхом проведення розрахунків, випробувань та обробки результатів експлуатації виробів, моделювання на ЕОМ (електронні обчислювальні машини), а також шляхом аналізу фізико-хімічних процесів, які впливають на надійність виробів. Розрахунки надійності ґрунтуються на тому, що при певній структурі виробу та певному законі розподілу ресурсу існують певні залежності між показниками надійності окремих елементів та загальною надійністю виробу. Для встановлення таких залежностей використовуються наступні методи: розв'язання рівнянь, складених на основі структурної схеми надійності (застосування послідовно-паралельних структур) або на основі логічних зв'язків між станами виробу (використання алгебри логіки); розв'язання диференціальних рівнянь, що описують процес переходу виробу з одного стану в інший (застосування графів станів); складання функцій, які описують стани складного виробу.

Зазвичай, розрахунки надійності проводяться на етапі проектування виробів з метою прогнозування очікуваної надійності даного варіанту виробу. Це дозволяє визначити найбільш відповідну конструкцію та методи забезпечення надійності, виявити можливі «слабкі місця», аргументовано призначити робочі режими, форму та порядок обслуговування виробу. Випробування на надійність проводяться на етапах розробки дослідного

зразка і серійного виробництва виробу. Існують такі види випробувань:

1) первинні, які визначають показники надійності;

2) контрольні, що спрямовані на контроль якості технологічного процесу з метою забезпечення заданого рівня надійності;

3) прискорені, під час яких використовуються фактори для прискорення процесу виникнення відмов;

4) неруйнівні, засновані на використанні методів дефектоскопії та інтроскопії, а також на аналізі непрямих ознак виникнення відмов. Моделювання на ЕОМ є найбільш ефективним способом аналізу надійності складних систем. Два широко використовувані алгоритми моделювання включають: перший, що базується на моделюванні фізичних процесів у досліджуваному об'єкті (оцінка надійності здійснюється за кількістю виходів параметрів об'єкту, що виходять за межі допустимого); другий, що ґрунтується на розв'язанні систем рівнянь, що описують стани досліджуваного об'єкту [12].

Оцінка надійності вивченого стану можлива за допомогою аналізу фізико-хімічних процесів, оскільки часто можна встановити взаємозв'язок між надійністю та характером цих процесів (таких як відношення міцності до навантаження, стійкість до зношування, наявність домішок у матеріалах, зміни електричних і магнітних властивостей, шумові ефекти і т. д.). Часто такий аналіз використовується для оцінки елементів радіоелектронної техніки.

Більшість користувачів систематично використовують комп'ютери, не замислюючись про можливість їхнього вимкнення і навіть неможливості подальшого включення. Часто виникають ситуації, коли зібраний або оновлений комп'ютер не запускається або раптово вимикається. У таких випадках важливо правильно визначити причину несправності, оскільки ремонт у деяких випадках може бути непотрібним [13]. Спочатку слід вивчити можливі фактори, які можуть викликати несправність.

Наприклад, пил та несприятливі кліматичні умови можуть погіршити стан компонентів персонального комп'ютера. Це може призвести до збоїв у роботі апаратного забезпечення через окислення контактів, потрапляння пилу (і, відповідно, статичної електрики) на мікросхеми та роз'єми, а також через температурні зміни.

Зазвичай всі ці несправності можуть бути результатом стрибків напруги, неправильної роботи блока живлення або недостатнього заземлення. У першу чергу рекомендується використовувати мережеві фільтри та заземлення комп'ютера. Якщо виявлено ознаки несправності, слід провести візуальний огляд та перевірити всі компоненти на відповідність нормам. Якщо очевидних ознак не виявлено, можна перевірити надійність підключення до живлення.

Якщо перевірка не принесла результатів, можна спробувати включити комп'ютер і перевірити роботу вентиляторів на блоку живлення та на кулері процесора. Якщо вентилятор не працює, а жорсткий диск не видав звичного звуку, можна припустити, що причиною є не справність блока живлення [14].

Наявність напруги на виході блоку живлення можна перевірити за допомогою тестера, провівши вимірювання напруги на контактах системної плати там, де вони з'єднані з блоком живлення. Щоб підтвердити це, можна підключити інший блок живлення і перевірити роботу інших компонентів комп’ютера. Хоча поломки моніторів відбуваються не так часто, варто перевірити, чи отримує він сигнали з відеоадаптера. Для цього потрібно застосувати осцилограф і перевірити наявність активних сигналів.

Комплекс для автоматичної діагностики комп'ютера включає в себе різноманітні програмні, мікропрограмні та апаратні рішення. Це включає системи тестування і функціональної перевірки [15]. У системах тестової діагностики результати передаються на діагностичний пристрій від засобів діагностики. У середніх і великих ЕОМ, як правило, використовуються спеціалізовані засоби діагностики. У мікро-ЕОМ частіше використовуються

вбудовані засоби, які подають тестові впливи на зовнішні пристрої для збору і обробки результатів. Процес діагностики включає кілька етапів (простих перевірок), кожен з яких характеризується наданням на пристрій або робочою напругою, що вимірюється з відповідного пристрою. Оскільки існує безліч можливих проблем, які можуть виникнути, не існує єдиного підходу до діагностики комп’ютера. Типові проблеми з апаратною частиною комп’ютера включають наступне:

- комп’ютер не вмикається;
- комп’ютер вмикається, але на екрані немає зображення;
- комп’ютер вмикається і видавати специфічні звуки («пищить»);
- помилки, пов’язані зі стартом BIOS;
- недіючі компоненти;
- перегрівання комп’ютера;
- несумісність обладнання;
- повільна робота системи.

Отже, очевидно, що проведення діагностики комп'ютерної системи можна здійснювати за допомогою фізичних зусиль людини. Проте, в якості альтернативи діє програмне забезпечення, яке наразі є досить поширеним серед активних користувачів комп'ютера [16]. Ці програмні засоби представляють собою набір команд, які надсилаються до кожного компонента комп'ютера і дають певний результат, який відображається у діалоговому вікні. Програмна система діагностики базується на вже введених даних про нормальну роботу компонентів комп'ютера, порівнюючи їх між собою, проводячи аналіз і відображаючи результати. Для отримання базової інформації про персональний комп’ютер можна використовувати стандартні інструменти операційної системи Windows. Найбільш доступним методом є використання Диспетчера завдань (рис. 1.2).

Для проведення діагностики комп'ютерних систем існує безліч способів, включаючи програмні, апаратні та навіть спеціальні веб-ресурси.

Проте загальні методи аналізу та діагностики залишаються однаковими для всіх цих засобів. Це включає обов'язкову перевірку складових системного блоку на наявність фізичних дефектів, а також використання програмних інструментів для ідентифікації похибок та несправностей, що можуть стосуватися як програмного коду системних додатків, так і характеристик апаратного забезпечення [17].

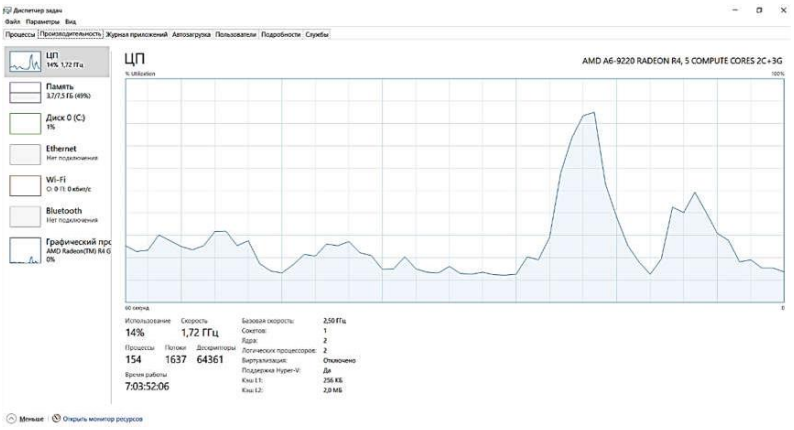

Рисунок 1.2 – Поточні характеристики комп’ютера

Аналіз комп'ютерної системи – це широко поширена і важлива тема для всіх користувачів комп'ютерів. Своєчасна діагностика дозволяє запобігти серйозним проблемам і уникнути необхідності у складних ремонтних роботах, які можуть забирати значні кошти. Тому при діагностиці важливо уважно перевіряти кожен компонент, незалежно від його значущості. Це особливо важливо при перевірці системних програм на наявність різноманітних вірусів, оскільки вони можуть приховуватися в найдрібніших файлах і призвести до серйозних проблем як у програмному, так і у апаратному забезпеченні. Часто такі пошкодження системи є дуже складними або навіть неможливими для відновлення. Для вирішення даної проблеми необхідно побудувати модель інтелектуального класифікатора, за допомогою якого буде проведена діагностика стану мережі та визначення причини збою з подальшими рекомендаціями. В якості простору ознак для проєктування нейромережевого забезпечення бізнес-процесів потрібно обрати показники, необхідні для оперативного визначення стану мережі.

Необхідно сформувати математичну постановку завдання, на основі якої буде проводитися створення нейромережевого забезпечення.

Математична постановка завдання виглядає наступним чином:

$$| y^k |= F | x^m | \quad (1.1)$$

Для вирішення поставленої задачі необхідно знайти відображення:

$$X_1 \subset \Re^L,\ X_n \subset \Re^G,\ Y \subset \Re^k, \quad (1.2)$$

де ($X_1 \rightarrow \ldots X n$) – 1, 2, … n – простір ознак,

F– способи визначення класів,

L, G – розмірність ознак в просторах 1, 2, …, n,

k – кількість досліджуваних класів (розмірність класів).

Для оцінки інформативності кожного з просторів ознак X важливо провести аналіз впливу відсутності або присутності конкретних просторів ознак на продуктивність моделі інтелектуального класифікатора. Це допоможе виключити простори ознак з низьким рівнем інформативності та забезпечити максимальну точність діагностики стану мережі.

Оцінка інформативності просторів ознак та побудова моделі на основі різних технологій діагностики стану мережі є складними процесами. В рамках магістерської роботи рекомендується зосередитися на побудові моделі інтелектуального класифікатора нейромережевого забезпечення, адаптованої для аналізу базових показників. Ця модель дозволить прогнозувати можливість збою та визначати його причину.

Після побудови базової моделі можна буде розглядати моделі з використанням більш широких просторів ознак, що базуються на більшій кількості показників діяльності мережі з урахуванням її топології, кількості входів, рівня небезпеки тощо. Важливо забезпечити, щоб результати цих моделей були однаково точними та продуктивними.

Розроблюваний проєкт повинен включати створення нейромережевого аналізатора інформаційної системи підприємства для аналізу причин збою в

мережі. Цей аналізатор допоможе підприємству оперативно реагувати на потенційні проблеми та забезпечить ефективне управління мережею.

Одним із перспективних напрямків розвитку та вдосконалення управління є розробка та впровадження передових інформаційних технологій на підприємстві, що включають у себе наступне:

1) Визначення функцій, які потрібно вирішувати для забезпечення надійного та якісного інформаційного забезпечення служб підприємства з метою прийняття рішень.

2) Визначення завдань, які необхідно вирішити для досягнення цих функцій, зокрема на першому етапі реалізації.

3) Визначення переліку якісних та кількісних показників інформації, необхідних для виконання завдань інформаційної системи.

4) Встановлення способів і методів, ґрунтуючись на яких або використовуючи які за допомогою різноманітних показників (якісних та кількісних), досягається вирішення поставлених завдань та визначення необхідних функцій для прийняття відповідних рішень.

1.4 Формування специфікації вимог до розроблюваного програмного забезпечення

У даному дослідженні пропонується методика, спрямована на вирішення проблем діагностики збоїв у комп'ютерній техніці. Вона базується на розробці експертної інтелектуальної системи, що використовує технологію на основі правил для виявлення таких збоїв. Замість того, щоб вручну перевіряти окремі компоненти, користувачі можуть просто ввести інформацію про своє комп'ютерне обладнання та симптоми, а потім система автоматично діагностує проблеми. Надаючи можливі причини збоїв та рекомендації щодо їх вирішення, експертна система допомагає користувачам у самостійній діагностиці та усуненні несправностей обладнання.

Правила роботи експертної системи сформульовані у формі тверджень «ЯКЩО-ТО», використовуючи конкретні категорії обладнання. Основна мета цієї системи - надати користувачам комп'ютерів можливість ефективно діагностувати та вирішувати проблеми з апаратним забезпеченням, що дозволяє їм вирішити невеликі неполадки або усунути несправності до звернення за професійною допомогою.

Висновки до розділу 1

В першому розділі кваліфікаційної роботи магістра проведено аналіз предметної сфери, пов’язаної з діагностикою збоїв у комп’ютерній мережі. В ході аналізу досліджено сучасний стан ринку комп’ютерних систем, розглянуто типові проблеми, з якими стикаються користувачі, а також визначено основні аспекти що впливають на ефективність діагностики та усунення несправностей обладнання.

Постановка завдання полягала у розробці ефективної експертної інтелектуальної системи, яка б могла автоматизувати процес діагностики збоїв у мережі. Специфікація вимог включала в себе опис функціональності системи, необхідність інтеграції з різними типами комп’ютерного обладнання, а також вимоги до швидкодії та точності діагностики.

З урахуванням проведеного аналізу та постановки завдання була розроблена концепція експертної системи, яка базується на технології на основі правил і забезпечує швидке та точне виявлення причин збоїв у комп’ютерах. Такий підхід дозволяє користувачам самостійно вирішувати більшість апаратних проблем без необхідності звертатися за професійною підтримкою.

2 РОЗРОБКА ПРОЄКТНИХ РІШЕНЬ ДЛЯ ДІАГНОСТИКИ ЗБОЇВ В МЕРЕЖІ

2.1 Формалізація процесу діагностики збоїв в мережі

Для діагностики збоїв існує потреба у визначенні архітектури мережі та основних її вузлів, у яких може відбуватись збій. З метою забезпечення спостережливості інформації в системах обміну інформацією використовуються спеціальні програми моніторингу. Програмне забезпечення моніторингу – це набір програмних засобів, які призначені для стеження за системами зв’язку. Вони дозволяють фіксувати активність користувачів та процесів, використовуючи ресурси, а також однозначно ідентифікувати їх учасників у конкретних подіях. Головна мета полягає у запобіганні можливим порушенням безпеки та відповідальності за певні дії.

Застосування програмного забезпечення моніторингу дозволяє автоматизувати процеси контролю за безпекою системи зв’язку, надаючи власнику (адміністратору безпеки) змогу:

– виявляти всі випадки несанкціонованого доступу до конфіденційної інформації, включаючи точний час та станцію мережі, з якої була здійснена спроба;

– локалізувати всі випадки спотворення або знищення інформації;

– встановлювати факти несанкціонованого встановлення програмного забезпечення;

– контролювати використання персональних комп’ютерів поза робочим часом та визначати їх мету;

– виявляти несанкціоноване використання модемів у локальній мережі шляхом аналізу запуску несанкціонованих програм;

– виявити випадки введення спеціальних слів і фраз на клавіатурі чи іншому кодонабірному пристрої, інші випадки спроби несанкціонованої авторизації в системі для доступу до вузлу зв’язку;

– виявити факти неправомірного використання пристроїв зв’язку;

– отримати надійну інформацію, що буде використана для розробки політики інформаційної безпеки компанії;

– контроль доступу до серверів, персональних комп’ютерів, інших пристроїв комунікацій;

– провести інформаційний аудит;

– проводити дослідження виявлених інцидентів;

– проводити дослідження, пов’язані з визначенням точності, ефективності та адекватності реакцій працівників на зовнішні дії;

– визначити завантаженість комунікаційних станцій;

– розробити механізми відновлення критичної інформації після збоїв системи;

– забезпечити спостережливість системи. Ця особливість, в залежності від якості виконання, допомагає у певній мірі контролювати відповідність працівників компанії встановленим правилам безпеки роботи на комп’ютерах та політики безпеки.

Існують основні методи захисту від атак: програмні, апаратні і хмарні.

Програмні рішення є найпопулярнішими на ринку. Вони складаються з набору інструментів для фільтрації трафіку, розроблених на основі власного досвіду розробника. Ці рішення прості у використанні, але ефективні лише проти незначних атак, таких як вандалізм.

Апаратні рішення включають створення розподіленої мережевої інфраструктури з великим запасом пропускної здатності. Їх використовують у великих мережевих структурах, таких як точки обміну трафіком, дата-центри і великі регіональні провайдери.

Хмарні рішення включають в себе мережеві структури з високою пропускною здатністю, де додані сервери для відсіювання шкідливого трафіку. Ця мережа поступово очищає трафік від шкідливих елементів, що призводить до зменшення їх кількості.

Серед основних способів забезпечення безпеки інформації на вузлах зв’язку слід виділяти [16; с. 67]:

1. Підвищення відношення сигнал/шум шляхом:

– збільшення потужності сигналу шляхом використання енергії, наприклад, через посилення сигналу на різних точках мережі з обслуговуванням чи без нього, однак це потребує значних витрат енергії або матеріалів;

– зменшення рівня шуму шляхом використання спеціальних ліній зв’язку з низьким рівнем власного шуму, наприклад, оптоволоконних кабелів, але це також потребує значних матеріальних витрат.

2. Використання методів захисту з використанням групових (мажоритарних) підходів базується на використанні від трьох до п’яти каналів зв’язку, які зазвичай розташовані фізично (часто географічно) в різних місцях, для передачі однієї і тієї ж інформації. Іноді також використовується багатократна передача цієї ж інформації по одному каналу зв’язку (від три до п’яти разів). У першому випадку це вимагає значних матеріальних витрат, тоді як у другому випадку пропускна здатність каналу зв’язку зменшується від трьох до п’яти разів. Отже, у системах передачі даних високої швидкості використання цих методів не завжди є доцільним.

3. Моніторинг цілісності інформаційних об’єктів, таких як програмні засоби та дані під час їх обробки та передачі, включаючи відновлення зруйнованої інформації, шляхом:

– застосування різноманітних завадостійких кодів для виявлення помилок у прийнятій інформації, що дозволяє реалізувати засоби виявлення спотворень у програмному, апаратному або програмно-апаратному забезпеченні;

– застосування різноманітних корекційних кодів, що відповідають на загрози, для реалізації засобів виявлення та виправлення спотворень інформації у системі.

Для забезпечення контролю цілісності інформаційних об’єктів та відновлення пошкодженої інформації, до складу інформації, яка захищається, введено концепцію надмірної інформації, що є ознакою цілісності або контрольною ознакою. Ця додаткова інформація представляє собою унікальний образ, який відображає цілісність даних, процедура формування якого відома, і який з дуже високою ймовірністю відповідає інформації, що захищається.

Існує багато методів захисту від помилок, які можна розділити на три основні групи: групові методи, коригувальне кодування з урахуванням завад та методи захисту від помилок у системах передачі зі зворотним зв’язком.

2.2 Визначення логічної структури програмного забезпечення

Для створення набору актуальних причин неполадок доцільно застосовувати структуру з двох складових. Перша частина має описувати подію небажаного випадку та об’єкт, на якому вона відбулася, а друга - ситуацію, яка пояснює, чому саме така подія сталася (табл. 2.1).

Таблиця 2.1 – Формування множини можливих загроз.

Вид загрози	Об’єкти дій			
	Устаткування	Програми	Дані	Персонал
Витік інформації	Крадіжка носіїв, підключення, несанкціоноване використання ресурсів	Несанкціоноване копіювання, перехоплення	Крадіжка, копіювання, перехоплення	Передача відомостей про захист, розголошення
Порушення цілісності інформації	Підключення, модифікація, спеціальні вкладення, зміна режимів, несанкціоноване використання ресурсів	Впровадження «Троянських коней» та «жучків»	Спотворення, модифікація	Вербування, підкуп персоналу
Порушення працездатності системи	Зміна режимів, виведення з ладу, руйнування	Спотворення, вилучення, підміна	Видалення, спотворення	Звільнення з посади, фізичне усунення

Зараз використовуються два типи систем виявлення: системи виявлення аномалій і системи виявлення особливостей. Однак основним недоліком систем виявлення функцій є те, що вони спеціально розроблені для виявлення конкретних типів атак, як правило, тих, які вважалися найнебезпечнішими на момент розробки системи. Це може стати проблемою,

коли виникають нові атаки або коли відбуваються зміни в параметрах руху, оскільки процес виявлення потребує переадресації та вирішення ще раз. Системи виявлення аномалій у мережах використовують різні припущення про характеристики нормального інтернет-трафіку через його складність моделювання. Одним із таких припущень є статистична однорідність трафіку. Проте обсяг комп'ютерних систем, до яких застосовні ці припущення, і конкретні умови, необхідні для їх реалізації, не були ретельно вивчені або окреслені. Отже, навіть незначні зміни в моделях трафіку або пропонованих послугах можуть вимагати перенавчання алгоритму виявлення. Щоб вирішити цю проблему, одним із потенційних рішень є застосування комплексного підходу до побудови надійної системи захисту від атак. Цей підхід включатиме різні компоненти, такі як системний моніторинг, запис і зберігання історії транзакцій, підтримку спеціального сховища для інтелектуального аналізу зловмисників та їхньої поведінки, а також формулювання ефективної стратегії протидії.

Крім того, буде створено окреме сховище для зберігання детальної інформації про роботу системи. Цей репозиторій слугуватиме цінним ресурсом для розуміння та аналізу функціонування системи, дозволяючи краще ідентифікувати потенційні вразливості та області, які потрібно покращити. Крім того, буде створено ще одне сховище для розміщення аналітичних компонентів, спеціально розроблених для виявлення загроз і виявлення ознак шкідливої діяльності. Ці компоненти використовуватимуть складні алгоритми та методи для аналізу зібраних даних і надання сповіщень і попереджень у реальному часі, коли виявлено потенційні загрози. Нарешті, буде застосовано низку надійних заходів для протидії та пом'якшення будь-яких атак, які можуть виникнути. Ці заходи включатимуть різні протоколи безпеки, методи шифрування, системи виявлення вторгнень і брандмауери, серед іншого. Використовуючи ці заходи, система захисту забезпечить цілісність і безпеку системи, захищаючи від будь-яких потенційних

порушень безпеки або несанкціонованого доступу. Висунуто пропозицію створити комплексну систему захисту, що складається з кількох ключових елементів. По-перше, агенти стеження будуть розгорнуті для моніторингу та відстеження різноманітних дій у системі. Ці агенти збиратимуть цінні дані, які далі оброблятимуться та зберігатимуться за допомогою передових засобів попередньої обробки та зберігання. Підсумовуючи, запропонована система захисту використовуватиме комплексний підхід, що включає агенти відстеження, заходи попередньої обробки та зберігання, сховища для інформації про роботу системи та виявлення загроз, а також потужні заходи проти атак. Завдяки впровадженню цих елементів система буде обладнана для ефективного захисту від зловмисних дій і забезпечення загальної безпеки та стабільності системи [2; с. 572].

Перший етап — це відстеження трафіку, що передбачає захоплення пакетів для оцінки обсягу потоку даних, складу трафіку та активності користувача. Щоб виконати це завдання, необхідно розробити алгоритми, які можуть визначати оптимальну кількість і частоту захоплення пакетів на основі таких факторів, як завантаження каналу та інших відповідних параметрів. Важливо знайти баланс, тому що якщо пакети перехоплюються занадто часто, це може перешкоджати безперебійному потоку трафіку. І навпаки, якщо пакети перехоплюються через постійні проміжки часу, це може призвести до «сліпих зон», де певні дані не повідомляються або не враховуються. [2; с. 572].

Початковий крок у процесі передбачає попередню обробку перехоплених пакетів з подальшою оцінкою найбільш критичних загроз і збереженням зібраної інформації. Оскільки цей етап потребує швидкої оцінки з використанням мінімальних ресурсів, доцільно використовувати прості та адаптовані порогові значення або, якщо необхідно, послідовний CUSUM.

Важливим є процес аналізу даних під час їх завантаження в пам’ять, виявлення потенційних атак і оцінка рівня загрози. Коли інформацію буде збережено в сховищі, дуже важливо провести ретельну оцінку, щоб визначити можливі ризики. Щоб досягти цього, рекомендується використовувати багатоканальний CUSUM і алгоритми ковзного середнього, які були описані раніше. Ці алгоритми забезпечують надійний і ефективний засіб аналізу даних і виявлення будь-яких потенційних загроз або вразливостей. Використовуючи ці методи, організації можуть забезпечити безпеку та цілісність своїх даних і приймати обґрунтовані рішення для пом’якшення будь-яких ризиків, які можуть виникнути.

Аналіз фонових даних виконується на регулярній основі або за заздалегідь визначеним розкладом, щоб ідентифікувати різні типи атак, як-от спроби сканування, атаки деградації та імпульсні атаки. Оскільки ці атаки вважаються менш шкідливими, тепер необхідно провести більш детальний аналіз. Це передбачає використання методів аналізу даних, інтелектуальних систем правил, нейронних мереж та інших передових методів.

У процесі виявлення атаки важливо приймати рішення на основі певних порогових значень. Якщо ці попередньо визначені значення перевищено на будь-якому з попередніх етапів, це вказує на потенційну загрозу атаки. Отже, стає обов’язковим створення експертної системи, яка може оцінити рівень загрози та остаточно визначити, чи відбувається атака.

Оцінка ризику, вибір моделі, перевірка та пошук стратегії є ключовими кроками у визначенні відповідних контрзаходів у разі виявлення атаки. Ефективність цих контрзаходів може змінюватися залежно від типу та характеристик атаки. Отже, стратегію контрзаходів, також відому як «стратегія» контрзаходів, необхідно відповідним чином адаптувати. На стратегію впливає якість коригувальних дій, наприклад рівень обслуговування, що надається зареєстрованим користувачам. Щоб розробити потенційні стратегії, використовується аналітичне моделювання, щоб

зрозуміти взаємодію між зловмисником і агентами захисту. Завдяки дослідженню аналітичних моделей можна оцінити ефективність контрзаходів і передбачити їх можливі наслідки. Конфліктна взаємодія між зловмисником і системою захисту керує процесом, подібним до гри, при цьому навантаження системи є основним фактором, на який впливають гравці. Це навантаження можна виміряти як загальне навантаження системи або навантаження на певні критичні вузли, такі як ЦП, оперативна пам’ять або мережеві канали [2; с. 573].

Далі необхідно провести оцінку, щоб визначити кількість і силу залучених нападників. Ця оцінка дозволяє зрозуміти потенційну величину та інтенсивність загрози. Вимірюючи можливості та ресурси зловмисників, стає можливим відповідним чином адаптувати стратегію протидії. Перший крок передбачає визначення конкретного типу динаміки, що діє в даній ситуації. Це передбачає розуміння природи конфлікту та глибинної динаміки, яка його стимулює. Отримавши чітке розуміння динаміки, стає можливим розробити цілеспрямовану та відповідну стратегію протидії. Після застосування контрзаходів система повинна постійно оцінювати ефективність стратегії шляхом вимірювання поточної загрози. Якщо атака продовжується або виникають нові загрози, необхідно відповідно переглянути стратегію. Цей ітеративний процес гарантує, що стратегія контрзаходів залишається адаптивною та чутливою до динаміки розвитку конфронтаційної ситуації. Підсумовуючи, розробка стратегії протидії вимагає комплексної оцінки конфронтаційної динамічної моделі. Ретельно враховуючи різні параметри та кроки, описані вище, стає можливим створити надійну та ефективну стратегію, яка ефективно пом’якшує та нейтралізує загрозу. Регулярна оцінка та перегляд стратегії забезпечує її постійну ефективність в умовах мінливих обставин. Крім того, необхідна оцінка рівня загрози. Це передбачає аналіз серйозності та негайності потенційної шкоди, завданої зловмисниками. Завдяки точному оцінюванню рівня загрози стає можливим визначити

пріоритети та ефективно розподілити ресурси для розробки заходів протидії. Після встановлення контрзаходів їх необхідно застосувати, а результати порівняти з початковими прогнозами. Це передбачає вжиття заходів та реалізацію визначених протидій. Активно залучаючись до стратегії захисту, стає можливим оцінити її ефективність у пом’якшенні загрози. Порівняння фактичних результатів із початковими прогнозами дозволяє оцінити успіх стратегії. Визначення можливих контрзаходів і прогнозування їх наслідків є ще одним невід’ємним кроком у створенні ефективної стратегії. Розглядаючи різні захисні заходи та їхні потенційні результати, стає можливим приймати обґрунтовані рішення щодо того, які контрзаходи застосовувати. Цей прогнозний аналіз гарантує, що вибрані контрзаходи відповідають бажаним цілям і завданням стратегії [2; с. 573].

Отже, ми розробили попередні вимоги до структури та компонентів програмного засобу системи діагностики причин збоїв у мережі.

2.3 Вибір моделей та розробка алгоритму функціонування системи діагностики

Неможливість забезпечити в процесі функціонування розподіленої інформаційної мережі установи її абсолютну захищеність навіть при відсутності злочинних впливів змушує шукати додаткові методи і засоби підвищення безпеки функціонування інформаційної мережі на етапі експлуатації. Один із способів – оперативне виявлення дефектів в програмах та випадків викривлення даних за допомогою введення часової, інформаційної та програмної надмірності. Ці типи надмірності також можуть бути використані для швидкого відновлення пошкоджених програм і даних, а також для запобігання ризикам, які загрожують безпеці мережі.

Розглянемо загальну концепцію захисту розподіленої інформаційної мережі і алгоритм захисту інформації в нашій системі. Перш за все, необхідно визначити рівень загроз для створення системи захисту об'єкта.

Для нашої системи існує загроза або несанкціонованого доступу до системи з боку працівників або через зовнішні хакерські атаки, або існує загроза фізичного викрадення або пошкодження фізичних носіїв інформації.

Все вищевикладене зображено на рис. 2.1. у вигляді зв’язків між елементами можливих загроз та засобами захисту від даних видів загрози доступу до інформації та її пошкодження.

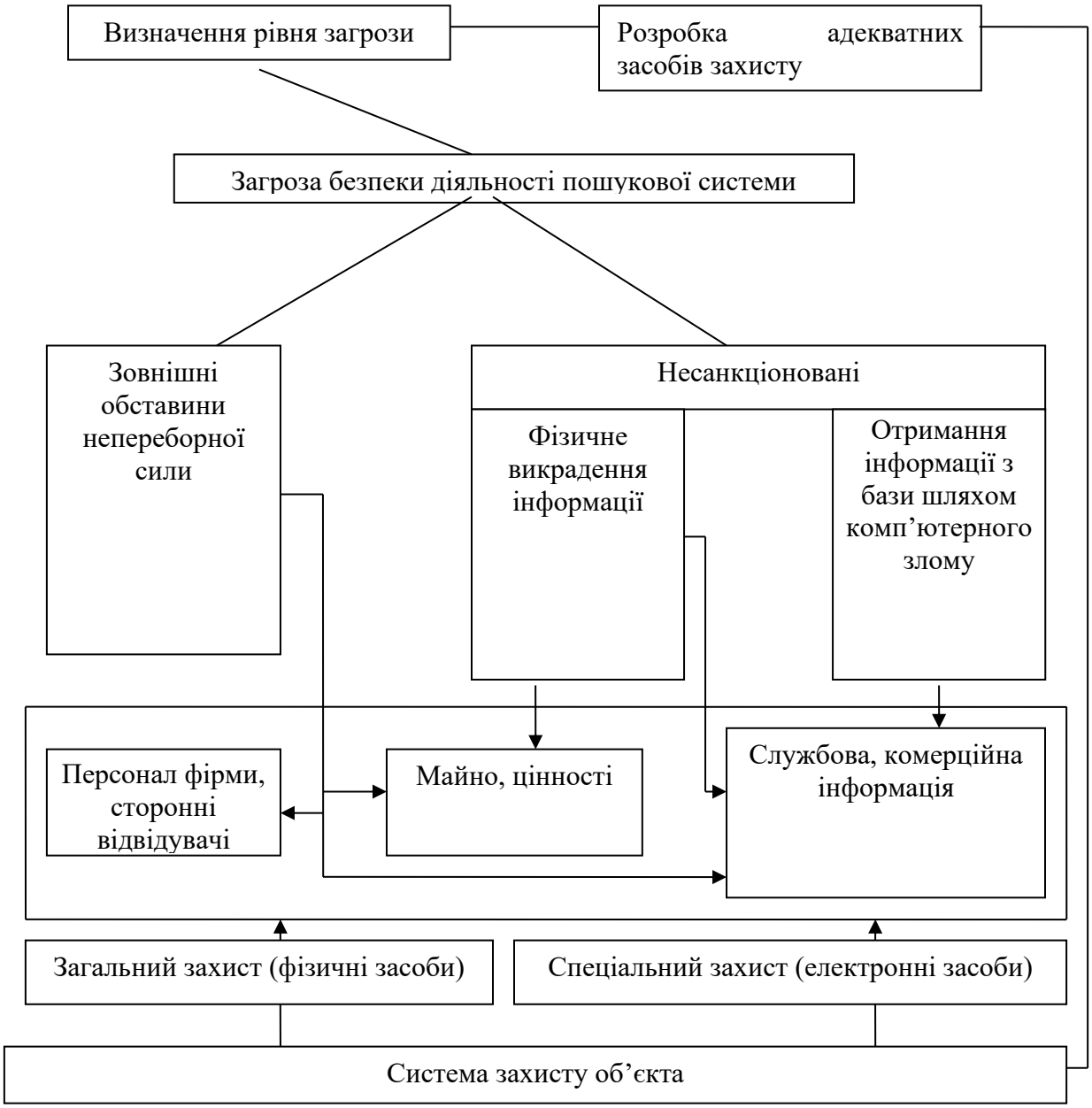

Рисунок 2.1 – Концепція захисту даних розподіленої ІС

У блок-схемі кожному типу операцій (введення-виведення даних, обчислення виразів, перевірка умов, керування повторенням дій, завершення обробки тощо) відповідає один або декілька блочних символів у формі плоских геометричних фігур, всередині яких міститься текст або формула, що пояснює здійснювані операції. Лінії зв’язку між блоками вказують на послідовність виконання операцій. Завдяки такій блок-схемі надається опис нечіткого алгоритму аналізу стану комп’ютерної системи (рис. 2.2), що є простим, і водночас інформативним та зрозумілим.

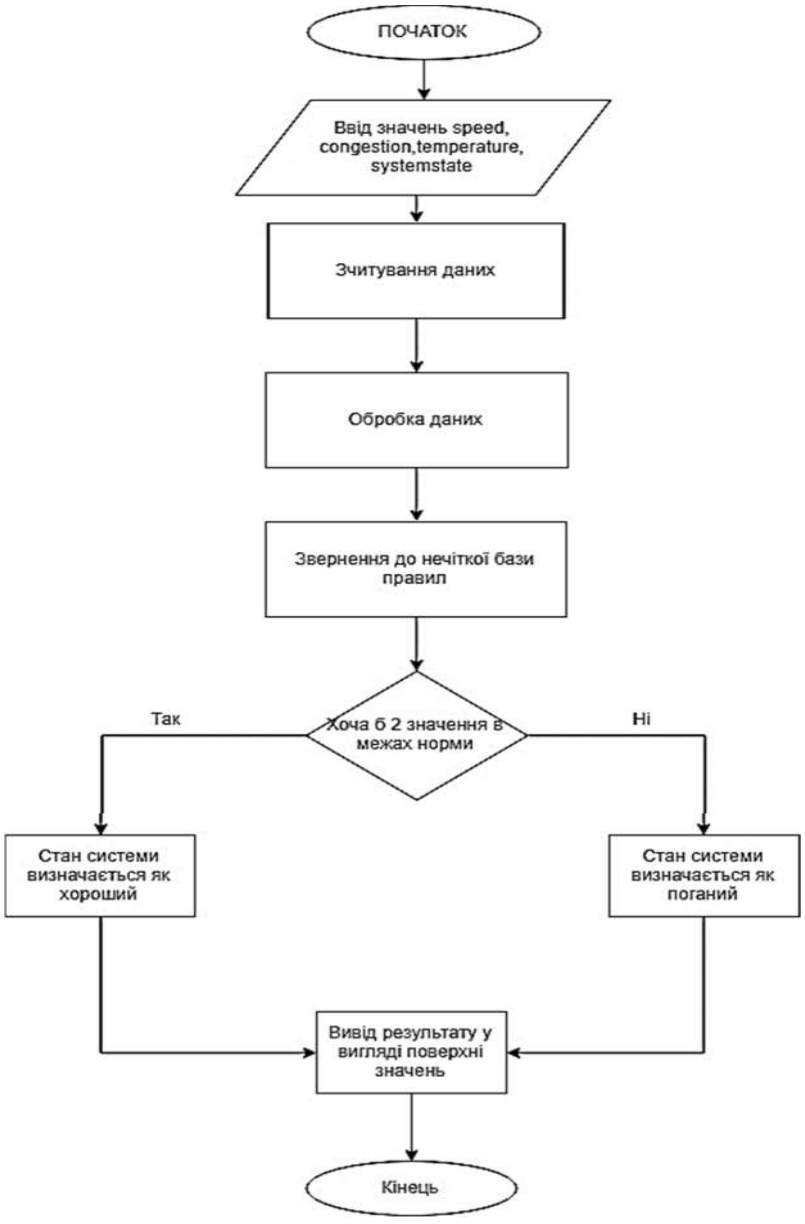

Рисунок 2.2 – Алгоритм аналізу стану комп’ютерної системи

Найпоширеніші способи подання алгоритмів включають звичайну мову, псевдокод (напівформалізований опис алгоритму), графічну форму та програмну реалізацію (текст на алгоритмічній мові програмування).

Аналізуючи ключові елементи розробленого алгоритму, можна зробити висновок, що його робота логічно та послідовно організована. Враховані всі важливі операції, необхідні для успішної реалізації розробленої системи.

Задачею дослідження є створення універсальної моделі оцінки вразливостей компонентів мережі. Ця модель має враховувати існуючі механізми виникнення збоїв, пропонувати заходи для їх нейтралізації, виявляти спроби несанкціонованого доступу та виникнення збоїв і вживати відповідних заходів для їх запобігання. Таким чином, ми дослідили основні типи загроз, методи та моделі ідентифікації причин збоїв в мережі. Результати нашого дослідження показали, що більшість рекомендацій, запропонованих іншими науковцями, не є достатньо систематизованими для ефективного аналізу причин збоїв. Цей висновок надихнув нас сформулювати та конкретизувати задачі дослідження для створення більш універсальних моделей діагностики мережі.

Досліджено завдання та процеси, які відводяться системі діагностики причин неполадок для розподіленої інформаційної мережі установи. Система повинна систематично перевіряти можливі загрози шляхом виявлення аномальної активності в будь-якому елементі і виконувати заплановані дії.

Висновки до розділу 2

В другому розділі здійснено формалізацію процесу діагностики збоїв в мережі. Визначено логічну структуру застосунку та алгоритм діагностування збоїв. Розроблено загальну концепцію захисту розподіленої інформаційної мережі.

3 ПРОЄКТУВАННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ ДІАГНОСТИКИ ЗБОЇВ В МЕРЕЖІ

3.1 Побудова UML-діаграм для проєкту

Щоб створити випадки використання, важливо ідентифікувати залучених осіб або організації. У разі розробки програмного забезпечення для виявлення вторгнень основним користувачем системи є адміністратор. Це програмне забезпечення спеціально розроблено, щоб допомогти адміністратору відстежувати та виявляти будь-які випадки збоїв у мережі. Таким чином, адміністратор є єдиним учасником цієї системи, як показано на рис. 3.1 [10, 18].

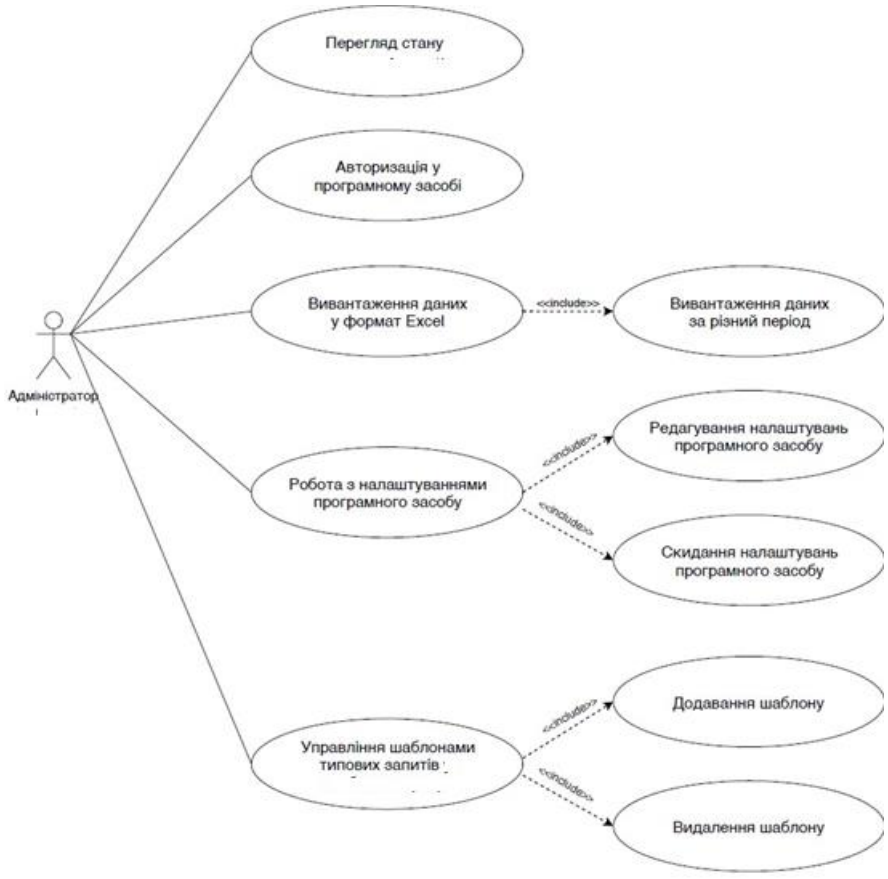

Рисунок 3.1 – Структурна схема варіантів використання

Детальний опис варіантів використання наведено в табл. 3.1 – 3.4.

Таблиця 3.1 – Авторизація для доступу до сторінки моніторингу мережі

Найменування	Авторизація для доступу до сторінки моніторингу мережі
Первинний актор	Системний адміністратор
Інші актори	Немає
Опис	Можливість авторизації для перегляду стану системи моніторингу мережі
Попередні умови	Відкрита сторінка моніторингу стану мережі
Вихідні умови	Збережена відкрита сторінка моніторингу стану мережі

Таблиця 3.2 – Конфігурація програмного забезпечення для діагностики збоїв

Найменування	Конфігурація програмного забезпечення для діагностики збоїв
Первинний актор	Системний адміністратор
Інші актори	Немає
Опис	Можливість налаштувати програмне забезпечення
Попередні умови	Створена відкрита сторінка налаштувань програмного забезпечення, а також авторизація адміністратора
Вихідні умови	Прийняті налаштування програмного забезпечення

Таблиця 3.3 – Експорт даних про причини збоїв у форматі Excel

Найменування	Експорт даних про причини збоїв у форматі Excel
Первинний актор	Системний адміністратор
Інші актори	Немає
Опис	Можливість експортувати дані про виявлені причини збоїв у форматі Excel
Попередні умови	Відкрита сторінка перегляду виявлених причин
Вихідні умови	Сформований файл у форматі Excel, що містить дані про причини збоїв

Таблиця 3.4 – Моніторинг стану мережі

Найменування	Моніторинг стану мережі
Первинний актор	Системний адміністратор
Інші актори	Немає
Опис	Можливість відстежувати стан мережі
Попередні умови	Сторінка для перегляду стану мережі відкрита, а також адміністратор авторизований
Вихідні умови	Відображення актуального стану мережі

Логічна структура складається з різних функціональних і логічних модулів, які складаються з процедур і об'єктів, які служать стандартизованими моделями для додатків баз даних. Компоненти цих модулів включають у себе форми, таблиці бази даних, вікна для перегляду, звіти, запити та інші. Крім того, логічна структура включає в себе унікальні програмні блоки, які призначені для автоматизації конкретних функцій або завдань у предметній області, що вивчається [19].

Схему класів програмного забезпечення, яка відображена на рис. 3.2, представлено за допомогою відповідної діаграми UML [10, 20].

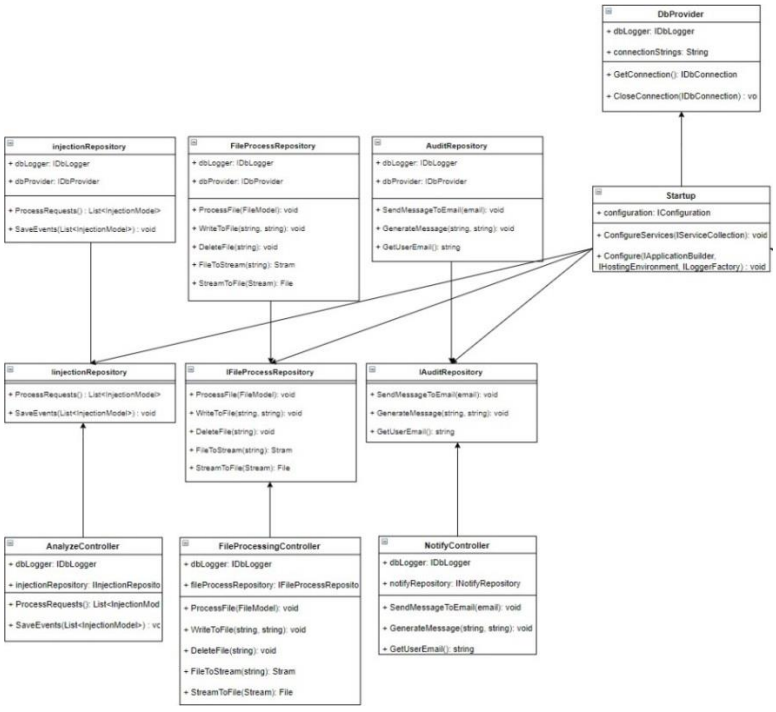

Рисунок 3.2 – Структурна схема класів

3.2 Проєктування інтерфейсу програми

Для реалізації нечіткої системи на основі розробленого алгоритму необхідно спочатку визначити вхідні параметри і бажаний вихідний результат. У нашому випадку вхідними даними є такі значення:

- температура;
- швидкодія;
- навантаженість.

Отриманий вихідний стан системи прямо відбувається від вхідних даних. Система нечіткого виводу створена для узагальнення емпіричних знань або експертних думок у визначеній сфері проблем. Правила висновку в системах нечіткого виведення формулюються за допомогою нечітких лінгвістичних висловлювань. База правил нечітких висновків - це остаточний комплект правил, які використовують лінгвістичні змінні, що узгоджені між собою. Часто базу правил подають у вигляді певної структурованої текстової форми. Загальний вигляд розробленої нечіткої системи зображено на рисунку 3.3.

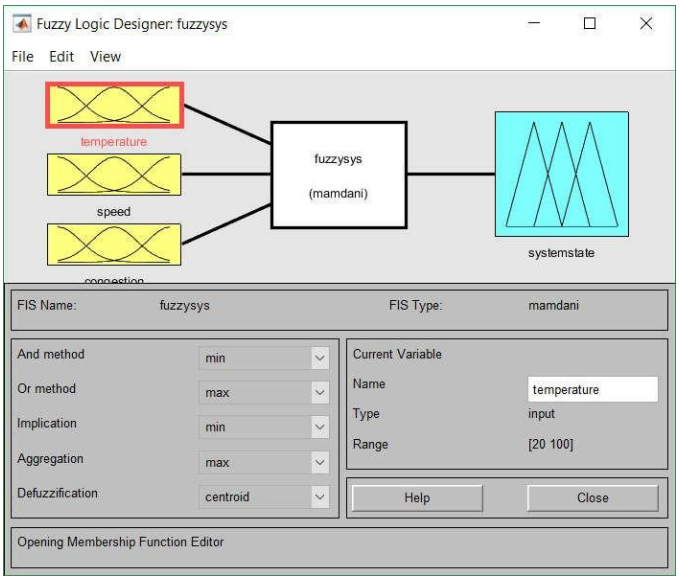

Рисунок 3.3 – Входи розробленої системи діагностики причин збоїв

Для подальшої роботи з нечітким виводом необхідно створити функції належності для кожного з входів (температура, швидкодія, навантаженість) і виходу (загальний стан системи). Обрано функції типу «gbellmf» для входів і «trimf» для виходу. Межі значень встановлені згідно з вказаними діапазонами.

Функції належності для температури:

1. Діапазон: від 20˚C до 100˚C
2. Температура в нормі: gbellmf(20, 35, 70)
3. Підвищена температура: gbellmf(35, 70, 100)
4. Висока температура: gbellmf(70, 85, 100)

Функції належності для швидкодії:

1. Діапазон: від 0 до 1
2. Висока швидкодія: gbellmf(0, 0, 0.5)
3. Нормальна швидкодія: gbellmf(0.4, 0.5, 0.7)
4. Низька швидкодія: gbellmf(0.6, 1, 1)

Функції належності для навантаженості:

1. Діапазон: від 0 до 200
2. Низька навантаженість: gbellmf(0, 0, 80)
3. Середня навантаженість: gbellmf(70, 100, 140)
4. Висока навантаженість: gbellmf(130, 160, 200)

Функція належності для загального стану системи:

1. Діапазон: від 0 до 10
2. Відмінний стан: trimf(0, 0, 3)
3. Нормальний стан: trimf(3, 5, 7)
4. Поганий стан: trimf(7, 10, 10)

Функції приналежності температури, швидкодії, завантаженості, а також кінцевого стану системи зображені відповідно на рисунках 3.4 – 3.7.

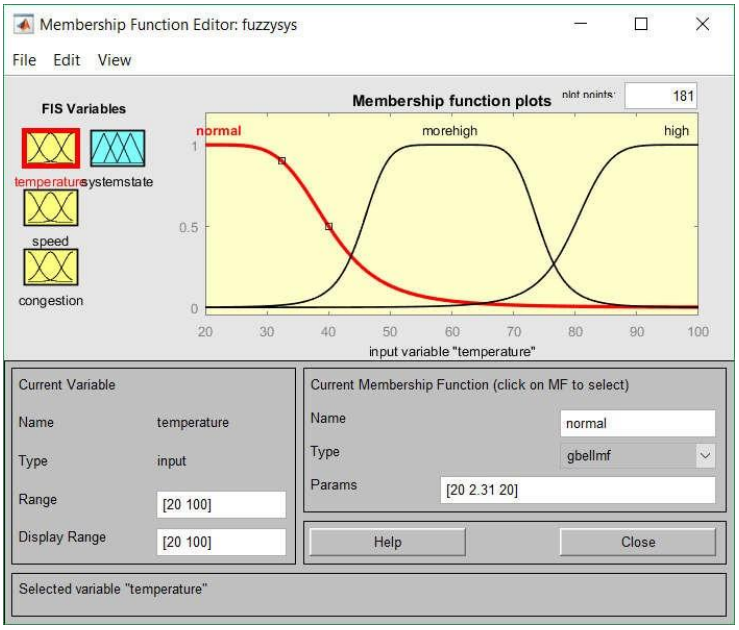

Рисунок 3.4 – Функції приналежності температури процесора

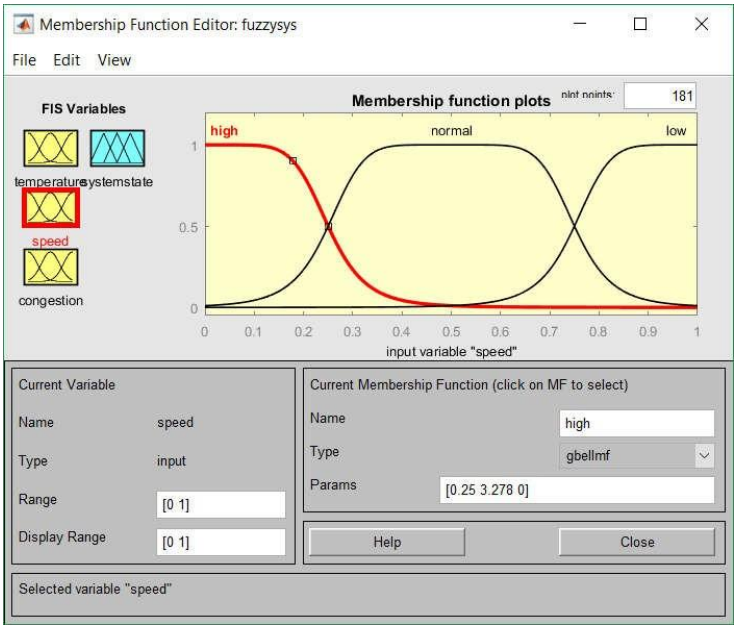

Рисунок 3.5 – Функції приналежності швидкодії мережі

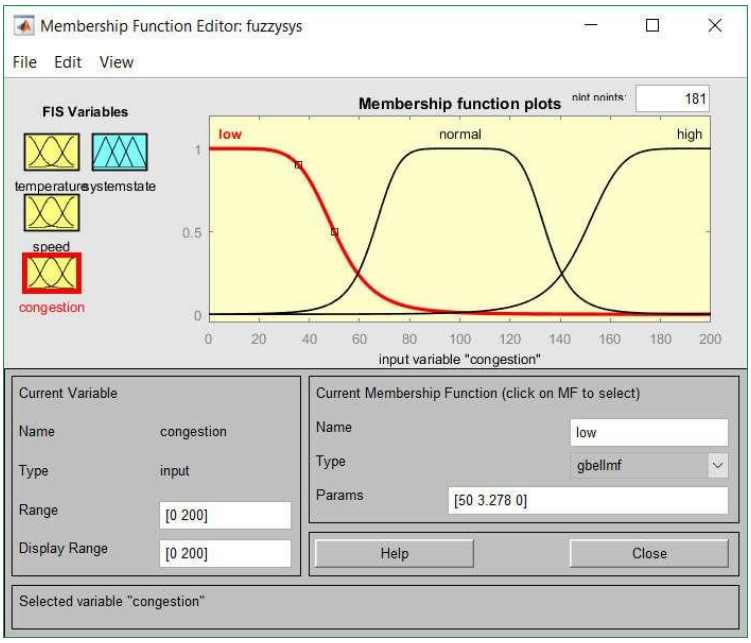

Рисунок 3.6 – Функції приналежності завантаженості мережі

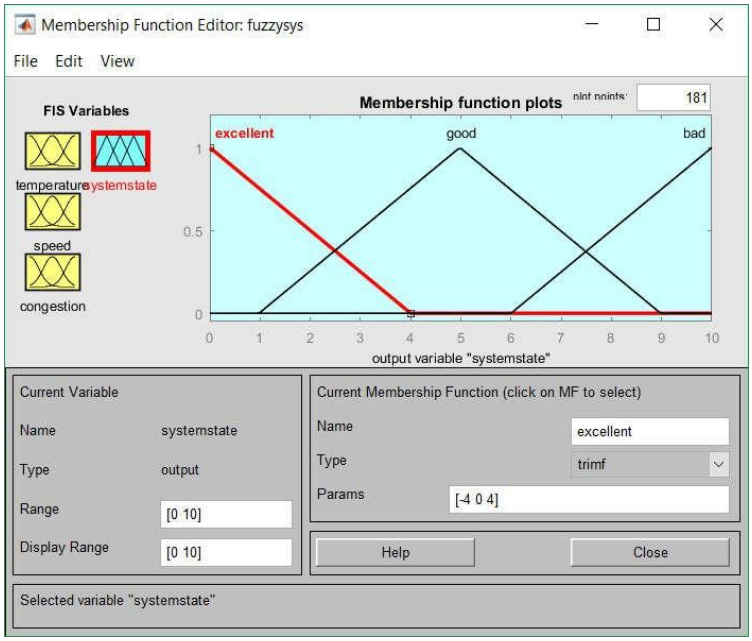

Рисунок 3.7 – Функції приналежності стану системи

Розроблена база нечітких правил включає сукупність правил типу «якщо – то», які встановлюють зв’язок між вхідними та вихідними даними об'єкта дослідження.

Система правил нечіткого виводу призначена для організації емпіричних даних чи експертної інформації в конкретній сфері. У таких системах використовуються правила нечітких продукцій, де умови й

висновки формулюються через терміни нечітких лінгвістичних висловлювань. [38].

Всі вхідні змінні мають по три нечітких стани, а також один стан «none», коли значення вхідної змінної не задане системою. Проте випадок, коли значення всіх вхідних змінних не задані, на практиці неможливий.

Для перевірки ефективності нечіткої системи надано поверхні значень зміни стану системи на рисунках 3.8 та 3.9.

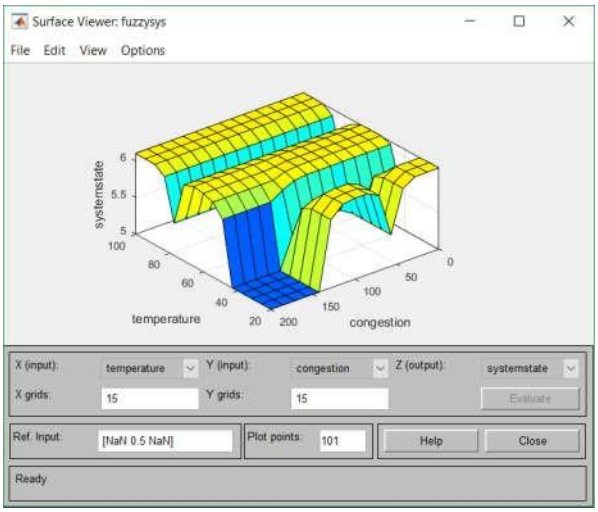

Рисунок 3.8 – Поверхня значень стану системи залежно відвідношення температури до навантаженості

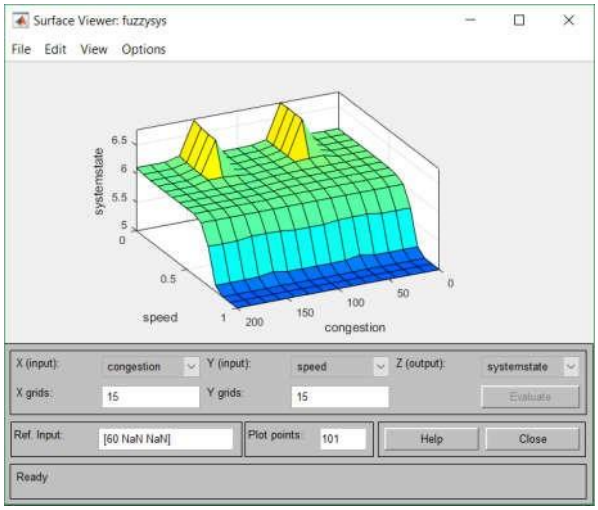

Рисунок 3.9 – Поверхня значень стану системи залежно відвідношення швидкодії до навантаженості

На рисунку 3.10 зображено нечіткий вивід моделі вибору стану комп’ютерної системи, що побудований за заданими правилами з поточними значеннями змінних.

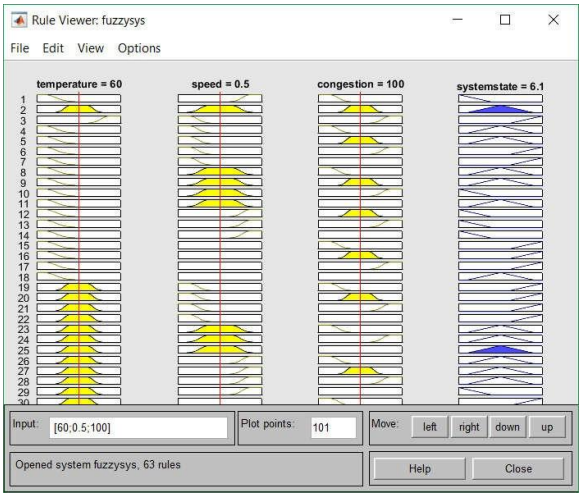

Рисунок 3.10 – Графічне зображення правил нечіткої системи

3.3 Проєктування структури автоматизованої системи діагностики збоїв в мережі

База правил систем нечіткого виводу спрямована на формалізацію практичних знань чи експертної інформації в конкретній області проблем. У таких системах використовуються правила нечітких продукцій, де умови і висновки формулюються за допомогою термінів нечітких лінгвістичних висловлювань. Кожне правило складається з умов і висновків, які у свою чергу є нечіткими висловлюваннями. Функція належності визначається на нечіткій множині, і її значення можна отримати за допомогою методу getValue(). Даний метод визначений в інтерфейсі FuzzySetIface.

Роботу системи побудовано на базі алгоритму Мамдані. Перший крок у функціонуванні системи - формування бази правил. Ця база складається з різних правил, кожному з яких приписаний свій ваговий коефіцієнт.

База правил може бути сформульована у такому форматі (для ілюстрації використовуються правила різних конструкцій):

RULE_1: IF «Condition_1» THEN «Conclusion_1» (F1) AND «Conclusion_2» (F2);

RULE_2: IF «Condition_2» AND «Condition_3» THEN «Conclusion_3» (F3);

RULE_n: IF «Condition_k» THEN «Conclusion (q–1)» (Fq–1) AND «Conclusion_q» (Fq);

Де Fq – це вагові коефіцієнти, які показують ступінь впевненості у правдивості отриманого висновку (i=1..q). За звичайних умов, ваговий коефіцієнт приймається рівним 1. Лінгвістичні змінні, що використовуються в умовах, називаються вхідними, а в результуючих даних – вихідними.

Використані позначення:

n – число правил нечітких продукцій (numberOfRules);

m – кількість вхідних змінних (numberOfInputVariables);

s – кількість вихідних змінних (numberOfOutputVariables);

k – загальна кількість підумов в базі правил (numberOfConditions);

q – загальне число висновків в базі правил (numberOfConclusions).

Другий етап у цьому процесі – етап фазифікації вхідних змінних, часто відомий як перетворення до нечіткості. На вхід надходять сформована база правил і масив вхідних даних $A = \{a_1, ..., a_m\}$, де m – кількість змінних. Мета цього етапу полягає в тому, щоб отримати значення істинності для кожної підумови, яка випливає з побудованої бази правил. Це досягається наступним чином: для кожної підумови ми знаходимо відповідне значення $b_i = \mu(a_i)$. Таким чином отримуємо множину значень b_i ($i = 1 .. k$). Подальшими етапами є агрегація підумов, активація та накопичення висновків. Агрегація підумов має на меті визначення рівня вірності передумов для кожного правила в системі нечіткого висновку. Формально це може бути виражено таким чином:

$$c_j = \min \{ b_i \},$$

де $j = 1..n$; i – число з множини номерів підумови в яких бере участь j –а вхідна змінна.

У процесі активізації та акумуляції висновків відбувається перехід від умов до підумов. Для кожного підвисновку знаходиться ступінь істинності $d_i = c_i * F_i$, де $i = 1..q$. Потім, знову ж кожному i –му підвисновку, спіставляється множина D_i з новою функцією приналежності. Значення «мінімум з di» визначається як результат порівняння значень di та функції належності терму з підвисновку [26]. Мета цієї фази полягає в створенні нечітких множин для кожної зі змінних. Це досягається шляхом порівняння і-тої вихідної змінної з об'єднанням множин $E_i = UD_j$, де j – номери підвисновків, в яких бере участь i –та вихідна змінна ($i = 1..s$). Злиття двох нечітких множин утворює третю нечітку множину з відповідною функцією належності:

$$\mu'_i(x) = \max\{\mu_1(x), \mu_2(x)\}, \text{ де } \mu_1(x), \mu_2(x).$$

Останнім кроком є процес дефазифікації вихідних змінних. Головна мета полягає в тому, щоб отримати чітке числове значення для кожної з лінгвістичних змінних. Цей процес можна формалізувати наступним чином. Розглядається і–та вихідна змінна множина Е, на якій ця змінна визначена – E_i ($i = 1..s$). Потім за допомогою методу дефазифікації розраховується підсумкове кількісне значення кожної вихідної змінної [2]. У всіх вхідних параметрах є по три різних стани, а також один стан "none", коли значення вхідної змінної не встановлене. Однак в реальних умовах малоймовірною є ситуація, коли всі вхідні змінні залишаються без визначення значень.

Висновки до розділу 3

В третьому розділі здійснено проєктування автоматизованої системи діагностування збоїв в мережі. Також здійснено специфікацію вимог, розроблено діаграму варіантів використання та діаграму класів.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ ДІАГНОСТИКИ ЗБОЇВ В МЕРЕЖІ

4.1 Опис середовища реалізації програми

1. Реалізація створення нечіткої системи буде найдоцільнішою за допомогою використання такого програмного середовища, як Matlab.

2. Середовищем називають пакет програм для проведення числового аналізу. Одночасно, це є високопродуктивною мовою програмування для проведення технічних обчислень. Ця мова використовується в пакеті Matlab відповідно. Ця платформа інтегрує обчислення, візуалізацію та програмування у зручному середовищі, де завдання та їх розв'язки представлені за допомогою знайомих математичних символів [40]. Програмне середовище Matlab найчастіше використовується для здійснення наступних функцій:

- вивчення та дослідження даних, проведення відповідного аналізу;
- візуалізація результатів досліджень;
- розробка алгоритмів;
- моделювання та прототипування;
- математика та обчислення;
- наукова та інженерна графіка;
- створення програмних застосунків, включаючи розробку графічного інтерфейсу для користувачів.

Matlab – це програмна система з інтерактивним інтерфейсом, головною особливістю якої є масив, що не потребує попереднього визначення розмірів. Це дозволяє ефективно вирішувати широкий спектр технічних обчислювальних задач, особливо тих, що вимагають матричних або векторних обчислень. В результаті значно зменшується час, потрібний для написання програм порівняно з традиційними скриптовими мовами, такими як C або Fortran. [41].

Назва Matlab скорочується від «Matrix Laboratory» – «Матрична Лабораторія». Ця система була спочатку розроблена для забезпечення студентам Університету Нью-Мексико простого доступу до матричного програмного забезпечення, що базується на проєктах Linpack та Eispack, які є передовими у світі програмними рішеннями для матричних обчислень.

Протягом перших декількох років Matlab розвивався за допомогою численної кількості користувачів. Цей пакет широко використовується серед університетів як стандартний засіб навчання для просунутих курсів з математики, техніки та інших наукових дисциплін. У промисловості також активно використовується Matlab як головний інструмент для розробки високопродуктивних досліджень та проведення аналізу [42].

Matlab – це серія програмних засобів для вирішення конкретних завдань, відомих як набори інструментів. Для більшості користувачів привабливість Matlab полягає в тому, що панелі інструментів дають можливість досліджувати та використовувати спеціалізовані технології. Панелі інструментів є пакетами функцій для Matlab (файли M), які розширюють можливості середовища Matlab для вирішення специфічних завдань. Ці пакети інструментів створені для різних областей, таких як нечітка логіка, обробка сигналів, нейронні мережі, системи управління, симуляція та інші. На рисунку 4.1 наведено типовий інтерфейс програмного середовища Matlab.

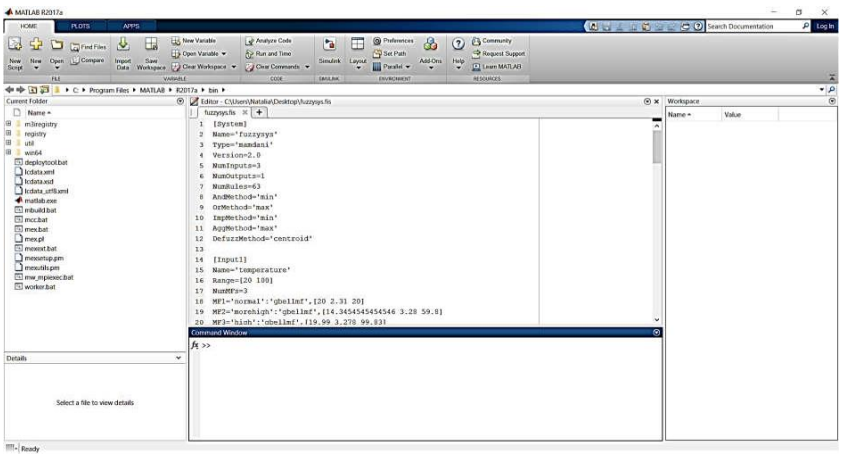

Рисунок 4.1 – Типовий інтерфейс середовища Matlab

У середовищі Matlab існує можливість налаштування спеціальних наборів інструментів (Toolbox з англійської), які додатково розширюють можливості цієї платформи. Такі набори складаються з різноманітних функцій, реалізованих на мові Matlab та призначених для вирішення конкретних класів завдань [43]. Підприємство Mathworks, що створила цю систему поставляє різноманітні набори інструментів, що використовуються в численних областях, в тому числі наступні:

– Цифрова обробка сигналів, зображень та даних. Приклади систем: Communication Toolbox, Wavelet Toolbox, Image Processing Toolbox, Filter Design Toolbox, DSP System Toolbox. Функціональність: вирішення широкого спектру завдань з обробки зображень, сигналів , проєктування систем зв'язку та цифрових фільтрів.

– Системи управління Приклади систем: Model Predictive Control Toolbox, Control Systems Toolbox, System Identification Toolbox, μ-Analysis and Synthesis Toolbox, Model–Based Calibration Toolbox, Robust Control Toolbox, LMI Control Toolbox. Функціональність: спрощення аналізу та синтезування динамічних систем.

– Фінансовий аналіз. Приклади систем: Financial Toolbox, Fixed-Income Toolbox, Financial Derivatives Toolbox, GARCH Toolbox, Datafeed Toolbox, Financial Time Series Toolbox. Функціональність: ефективний збір, обробка та передача різноманітної фінансової інформації в короткі терміни

– Аналізування і синтез географічних карт, включно з тривимірних Приклади систем: Mapping Toolbox. Збирання та аналізування експериментальних даних. Приклади систем: Image Acquisition Toolbox, Data Acquisition Toolbox, Link for Code Composer Studio, Instrument Control Toolbox. Функціональність: зберігання та обробка даних, що були отримані під час проведення експериментів, в тому числі в реальному часі. Додатково, перелічені програмами підтримують широкий спектр інженерного та наукового вимірювального обладнання.

– Візуалізація та уявлення даних. Приклади систем: Virtual Reality Toolbox. Функціональність: розробка інтерактивних віртуальних середовищ та візуалізація наукових даних шляхом використання технологій віртуальної реальності (VR) та мови VRML.

– Засоби розробки. Приклади систем: Matlab Builder for NET, Matlab Builder for Excel, Matlab Builder for COM, Filter Design HDL Coder, Matlab Compiler. Функціональність: розробка незалежних програм з середовища Matlab.

– Взаємодія з зовнішніми програмними продуктами. Приклади систем: Excel Link, Database Toolbox, , Link for ModelSim, Matlab Web Server, Matlab Report Generator. Функціональність: зберігання різноманітних даних таким чином, щоб їх можна було обробляти за допомогою інших програм.

– Бази даних. Приклади систем: Database Toolbox. Функціональність: опрацювання різних баз даних.

– Наукові та математичні пакети Приклади систем: RF Toolbox, Curve Fitting Toolbox, Fixed–Point Toolbox, OPC Toolbox, Fuzzy Logic Toolbox, Spline Toolbox, Bioinformatics Toolbox, Genetic Algorithm and Direct Search Toolbox, Partial Differential Equation Toolbox, Statistic Toolbox, Optimization Toolbox. Функціональність: вирішення широкого спектру наукових та інженерних завдань, що включають в себе, але не обмежуються вирішенням завдань в приватних похідних, розробкою генетичних алгоритмів, , оптимізацією систем, цілочисельними проблемами та іншими.

– Нейронні мережі. Приклади систем: Neural Network Toolbox. Функціональність: синтезія та аналіз різноманітних нейронних мереж.

– Нечітка логіка. Приклади систем: Fuzzy Logic Toolbox. Функціональність: побудова та аналіз нечітких множин.

– Символьні обчислення. Приклади систем: Symbolic Math Toolbox. Функціональність: реалізація символьних обчислень та їх інтеграція з символьним процесором Maple.

Узагальнення та аналіз вищенаведених даних дозволяє зробити висновок, що Matlab є програмним середовищем, що вирішує значний спектр проблем та допомагає в реалізації найрізноманітніших наукових рішень та ідей. Кожен користувач має можливість вибрати оптимальний набір функцій для свого проєкту, що забезпечить не лише ефективність вирішення завдань, але й спростить реалізацію через зрозумілий та простий інтерфейс. Це дозволяє значно зекономити час та ресурси, необхідні для розробки та впровадження проєкту [13]. Вищезазначене дає змогу стверджувати, що за допомогою програмного засобу Matlab можна вирішити широкий спектр актуальних завдань, оскільки з таким розмаїттям наборів функцій практично немає галузі чи наукової сфери, де б не було б доцільно використовувати Matlab.

Simulink – це інструмент для імітаційного моделювання, який пропонує можливість створювати динамічні моделі за допомогою блок-схем у вигляді графів, що містять різноманітні типи систем, такі як дискретні, безперервні і гібридні, системи з нелінійною та розривною динамікою [44].

Середовище Simulink є інтерактивним та дає користувачам доступ до готових бібліотек блоків для моделювання різних систем: електросилові, гідравлічні та механічні. Також воно надає можливість використовувати розширений модельно-орієнтований підхід при розробці різноманітних систем управління, пристроїв з цифрового зв'язку та засобів, що працюють в режимі реального часу.

Розширення Simulink має ряд додаткових пакетів, що значно поглиблюють можливості розв'язання широкого спектру завдань, починаючи від формування концепції моделі і закінчуючи тестуванням, верифікацією, генерацією коду та реалізацією на апаратному рівні. Simulink інтегрований у середовище MATLAB, що дозволяє використовувати вбудовані математичні алгоритми, потужні інструменти обробки даних і побудови графіків [45].

Simulink – це досить автономний інструмент, який відокремлений від Matlab, тому для його використання не потрібно мати глибоких знань у Matlab та інших компонентах. Однак варто зауважити, що доступ до функцій Matlab і інших інструментів залишається відкритим у Simulink і може бути використаним. Також, частина пакетів, що входять до складу Matlab, має вбудовані інструменти, які можна використовувати в Simulink (наприклад, LTI Viewer із пакету Control System Toolbox). Моделювання відбувається за допомогою використання стандартних моделей у пакеті Simulink. Під час моделювання в Simulink числові алгоритми компонентів систем автоматичного управління виконуються паралельно в кожен момент часу, що відомо як імітація часового потоку.

Simulink – це програмний інструмент, що дозволяє відобразити об'єкт аналізу у вигляді взаємопов'язаних блоків, що утворюють структурну схему. Після цього можна досліджувати його з поведінкової точки зору як у статичному, так і в динамічному режимі.

Розробник моделі в Simulink повинен засвоїти правила використання готових функціональних елементів, які використовуються для побудови моделі пристрою, а також, і це важливо підкреслити, «випробувальний стенд» – всю інфраструктуру, що складається, в тому числі, з джерел сигналів, вимірювальних пристроїв та засобів спостереження за процесами та особливостями, специфікаціями процесів, що розглядаються.

Блоки вводяться шляхом вибору з певного набору типових блоків, які включені у спеціальну бібліотеку. Ця бібліотека містить різноманітні блоки, що дозволяють моделювати різноманітні види лінійних, нелінійних, неперервних і дискретних елементів руху з численними змінними.

Інструмент SimuLink дозволяє проводити дослідження динамічних нелінійних систем шляхом моделювання у часі. Створення чисельної моделі системи, що вивчається здійснюється через графічне складання в спеціальному вікні схеми з’єднань елементарних візуальних блоків,

доступних у бібліотеках SimuLink. За своєю суттю, кожен блок, являє собою певну математичну програму. Зв'язки між різними програмами формуються через лінії з'єднання блоків, що дозволяють встановити послідовність викликів програм та передачу інформації між ними. Цей процес утворює програмну модель, яка зберігається у файлі з розширенням .mdl та відома як S-модель. Такий спосіб створення обчислювальних програм отримав назву візуального програмування.

Побудова моделей у пакеті SimuLink ґрунтується на використанні простої технології Drag-and-Drop. Під час створення S-моделі використовуються різні модулі, які можна перетягувати з бібліотеки SimuLink. Ці моделі можуть мати ієрархічну структуру, тобто складатися з інших моделей на більш низькому рівні. Кількість рівнів ієрархії необмежена. Під час моделювання можна спостерігати за процесами у системі за допомогою спеціальних блоків "оглядових вікон", які доступні у бібліотеці SimuLink. Крім того, користувач може розширити бібліотеку SimuLink, створивши свої власні блоки.

Використання SimuLink надзвичайно зручне при моделюванні систем, які складаються з різних функціональних пристроїв, поведінка яких описується залежностями, що відомі. Це дозволяє легко візуалізувати зв'язки між цими пристроями, які подані у вигляді блоків на блок-схемі S-моделі. Такий підхід суттєво спрощує програмний аналіз та синтез систем автоматичного керування [27, с. 4].

Щоб увімкнути систему нечіткого виведення в модулі Simulink, необхідно відкрити fuzblock командою fuzblock або за допомогою опції Fuzzy Logic Toolbox у браузері бібліотеки Simulink. Далі вибрати блок Fuzzy Logic Controller, двічі натиснуть і ввести ім’я файлу або ім’я змінної з цієї області, що відповідає системі нечіткого виводу, у діалоговому вікні, що з’явиться.

Якщо нечітка система має кілька входів, то в моделі Simulink вони повинні бути з'єднані один з одним до входу в нечіткий контролер. Подібним

чином, якщо нечітка система має кілька виходів, то виходи блоку будуть представлені однією мультиплексною лінією.

Залежно від конкретних задач дослідження, розрахункові моделі можуть бути розширені або спрощені шляхом врахування додаткових аспектів або введення припущень. Проте слід відмітити, що ускладнення розрахункової моделі зазвичай ускладнює процес математичного розв'язання і не завжди призводить до покращення точності результатів.

Для створення математичної моделі динамічної системи у формі структурної схеми скористаємося такими основними елементами бібліотеки Simulink (рис. 4.2).

Вигляд	Назва	Характеристика	Вигляд	Назва	Характеристика
1 Constant	Constant	Задає постійний за рівнем сигнал	du/dt Derivative	Derivative	Виконує числове диференціювання вхідного сигналу
× Product	Product	Виконує множення поточних значень сигналів	1 Gain	Gain	Виконує множення поточних сигналів на постійний коефіцієнт
1 In1	Inport	Утворює вхідний порт для підсистеми	1 Out1	Outport	Утворює вихідний порт для підсистеми
1/s Integrator	Integrator	Виконує інтегрування вхідного сигналу	+ +	Sum	Виконує додавання поточних значень сигналів
In1 Out1 Subsystem	Subsystem	Підсистема як фрагмент Simulink моделі, що оформлена у вигляді окремого блока	Signal 1 Signal Builder	Signal Bulder	Утворює лінійний сигнал довільної форми за допомогою графічного інтерфейсу користувача
Transport Delay	Transport Delay	Забезпечує затримку вхідного сигналу на заданий час	simout To Workspace	To Workspace	Записує дані, що надійшли на його вхід в робочу область MATLAB

Рисунок 4.2 – Елементи бібліотеки Simulink

Для роботи з інструментами нечіткої логіки призначено спеціальну надбудову Fuzzy Logic Designer, загальний вигляд якої зображено на рис. 4.3.

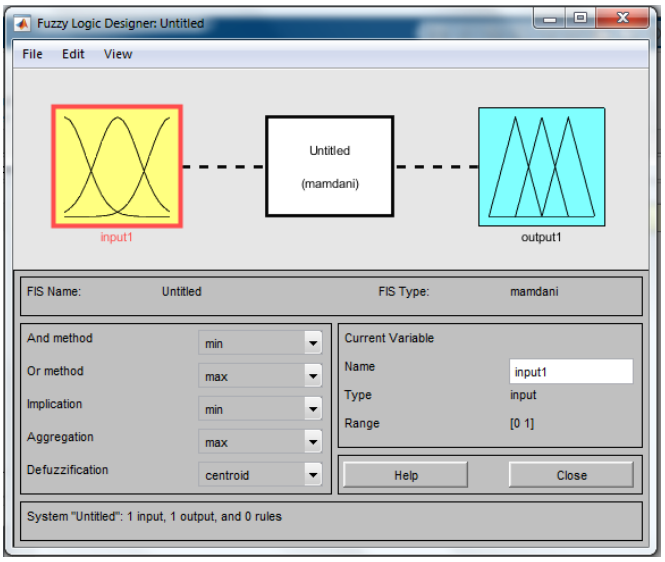

Рисунок 4.3 – Fuzzy Logic Designer: вікно створення нечіткої системи

Таким чином, ми обрали середовище розробки системи і переходимо до моделювання інформаційної системи з елементами нечіткої логіки.

4.2 Тестування розробленого програмного забезпечення

На рисунках 4.4, 4.5, 4.6 та 4.7 графічно показано значення вхідних змінних системи разом із вихідною змінною, яка відображає результат роботи і дозволяє зрозуміти загальний стан аналізованої системи.

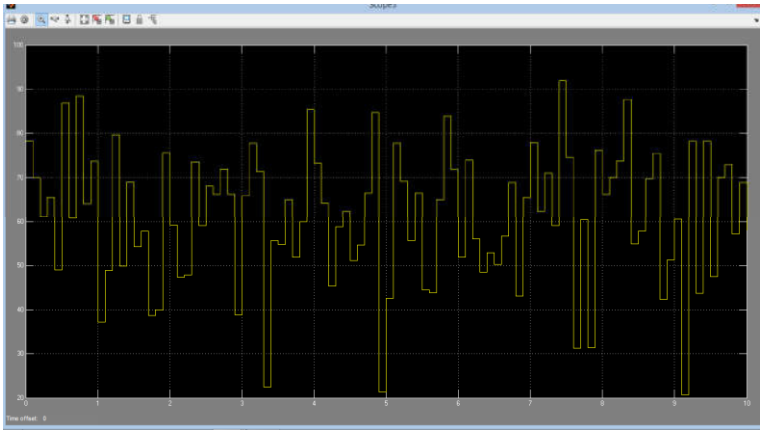

Рисунок 4.4 – Значення вхідної змінної температури процесора

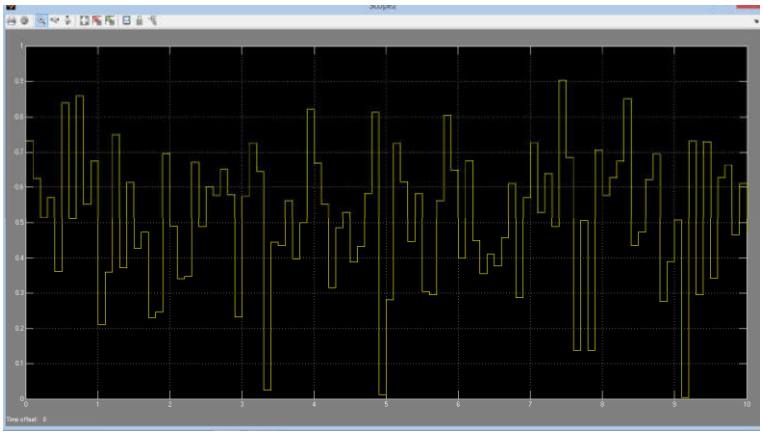

Рисунок 4.5 – Значення вхідної змінної швидкодії мережі

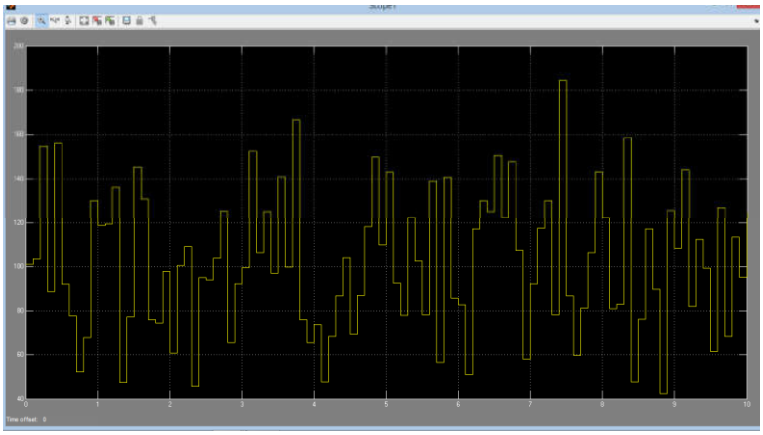

Рис. 4.6 – Значення вхідної змінної завантаженості на вузлах зв’язку

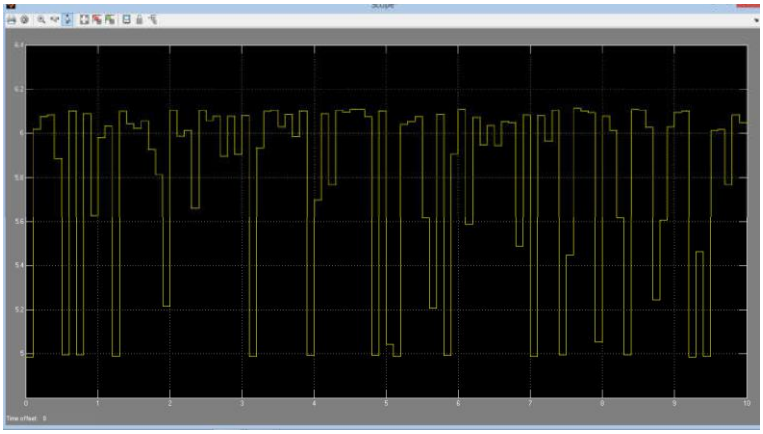

Рисунок 4.7 – Значення виходу нечіткого контролера

Таким чином, інформацію про роботу контролера можна легко отримати та відстежити. Достатньо того, щоб контролер зчитав дані з входів, обробив їх за допомогою бази правил, методів фазифікації та дефазифікації.

В результаті ми отримуємо чітко структуровані значення, які лежать в певних межах і дозволяють визначити кінцевий стан системи.

Також для підтвердження правильності роботи створеної нечіткої системи, представлено таблицю данихз результатами моделювання(таблиця 4.1).

Таблиця 4.1 – Результати моделювання нечіткої системи

Вхідні значення			Вихідне значення systemstate
temperature	Speed	congestion	
31.6	0.28	41.8	5.5
76.5	0.83	157	7
0	0.4	40	3.2
85	0	140	5.3
75	0.2	47	8
15	0.75	47	2.2
30.4	0.48	96	4.6
27.9	0.61	115.3	6.31
12	0.51	100.9	5.2
35	0.5	134.4	4.7

Отже, під час перевірки створеної нечіткої системи були аналізовані всі вхідні та вихідні дані. В результаті виявлено, що система працює належним чином і відповідає встановленим вимогам, у тому числі базі нечітких правил та відповідному алгоритму.

4.3 Розробка інструкції користувача

Посібник користувача розробленого програмного забезпечення містить ряд інструкцій щодо використання спільних ресурсів, входу та виходу з автоматичної системи, способів захисту персональних даних тощо.

Це буде виглядати так як показано нижче.

1. Увійдіть в систему. Вхід в систему здійснюється шляхом введення особистого логіна та пароля користувача. Якщо 3 рази поспіль буде введено неправильний пароль, доступ до системи буде заблоковано на 10 хвилин.

2. Робота в системі. Для роботи в системі виберіть один із запропонованих варіантів – перегляд, пошук, формування звіту тощо. В рамках вибраного варіанту користувачеві будуть запропоновані вбудовані функції для роботи з системою.

3. Введення інформації. Після входу в ІС користувач інформації потрапляє на домашню сторінку ІС, де знаходиться форма з полями для пошукового запиту, пропозиції варіантів та інші позиції. Користувач заповнює всі чи обрані поля і натискає кнопку «Далі» після завершення. В новому вікні видається результат пошуку причини збоїв, а також загальний стан системи згідно пошукового запиту.

4. Зміна пароля. Кожен користувач може змінити свій власний пароль. Доступ до цієї функції здійснюється через спеціальне меню, яке викликається після натискання на назву облікового запису у верхній правій частині екрана.

5. Закінчення роботи з ІС. Для завершення роботи з ІС та виходу з системи Користувач інформації повинен натиснути «Вихід» у меню, розташованому у верхньому правому куті вікна ІС.

Висновки до розділу 4

В четвертому розділі реалізовано інформаційну систему для діагностування збоїв у мережі. Продемонстровано принцип роботи ІС та складено керівництво користувача. Отже, процес розробки кваліфікаційної роботи магістра завершено.

ВИСНОВКИ

Моделі машинного навчання широко використовуються в галузі виявлення мережевої атаки. Методи, що застосовуються для генерації моделей машинного навчання, включають коди автомобілів, обмежені машини Больцмана, глибокі переконання та періодичні нейронні мережі. І навпаки, методи, що використовуються для диференціації, передусім включають згорткові нейронні мережі. Мета-навчання має на меті, щоб моделі могли набувати конкретного типу здатності до навчання, що дозволяє їм автономно вивчати мета-інформаційність. Ця метаінформація охоплює гіперпараметри, початкові параметри, структуру нейронної мережі та оптимізатор, які можна отримати зовні з процесу навчання моделі. Методи мета-тренувань часто покладаються на навчальні освітні моделі для різних завдань для вивчення метаінформації.

Навчання системи підвищує ефективність єдиного класифікатора, поєднуючи кілька основних класифікаторів, які, як правило, перевершують окремі класифікатори. Для класифікації незбалансованих зразків використовуються методи ансамблю і часто поєднуються з методами рівня даних. Цей підхід передбачає поєднання декількох простих класифікаторів за допомогою зваженої суми їх результатів класифікації для визначення остаточного результату класифікації. Зазвичай для класифікації незбалансованих зразків використовуються два методи, що використовують навчання ансамблю. У області аналізу мережевого трафіку методи онлайн - навчання зазвичай використовуються для класифікації мережевого трафіку, виявлення аномалії та аналізу журналу трафіку. Це дослідження в першу чергу зосереджується на проблемі виявлення аномалій мережі, яку можна вирішити, негайно виявляючи мережеві збої за допомогою онлайн -навчання.

Крім того, глибокі алгоритми навчання в Інтернеті все частіше адаптуються для задоволення потреб у режимі реального часу виявлення мережевого трафіку. Онлайн-навчання використовує масштабні послідовні

алгоритми зразка для створення прогнозних моделей, що робить його більш ефективним та придатним для обробки потокових даних із тимчасовими характеристиками. Онлайн-навчання ефективно фіксує зміни та тенденції даних, вирішуючи питання незбалансованого розподілу даних та полегшення навчання в режимі реального часу.

Традиційні методи глибокого навчання для ідентифікації мережевого збою в першу чергу зосереджуються на загальному виявленні атаки, що часто призводить до незбалансованих даних і згодом призводить до високої глобальної точності, але низької локальної точності класифікаторів машинного навчання. Це питання стало проблемою для дослідників через обмежену кількість навчальних зразків. Щоб подолати цей виклик, дослідники досліджували використання трансфертного навчання в ідентифікації мережевої атаки, особливо шляхом тонкої настройки, метричного навчання та мета-навчання. Поступове навчання фокусується на постійному оновленні моделей, використовуючи дані, що надходять у різні моменти часу. З появою великих даних генерування даних стає все більш важливим. Потоки даних характеризуються швидкісним потоком та динамічними змінами, що вимагає адаптивних методів додаткового навчання. Ці методи мають на меті здобути знання з нових даних, мінімізуючи вплив на раніше придбані знання.

Отже, аналіз можливостей нейронної мережі для виявлення причин та прогнозування збоїв показав, що періодичні нейронні мережі, засновані на довгострокових методах пам'яті, виявляють найвищу точність, оскільки вони не лише обробляють поточний стан для виявлення та реагування на аномалії.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Андон П.І., Ігнатенко О.П. Атаки на відмову в мережі Інтернет : опис проблеми та підходів до її вирішення. К. : Ін-т ПС, 2008. 52 с.

2. Андон П.І., Ігнатенко О.П. Протидія атакам на відмову в мережі інтернет: концепція підходу. Проблеми програмування. 2008. № 2-3. С. 564-574.

3. Антонюк П.Є. Класифікація ймовірних способів вчинення атак на інформацію як напрям протидії комп'ютерній злочинності. URL: http://www.nbuv.gov.ua/portal/Soc_Gum/bozk/19text/g1927.htm. (дата звернення 20.01.2024)

4. Бабенко Т.В. Дослідження ентропії мережевого трафіка як індикатора DDoS-атак. Науковий вісник НГУ. 2013. № 2. С. 86-89.

5. Багнюк Н.В., Мельник В.М., Клеха О.В. Види DDoS-атак та алгоритм виявлення DDoS-атак типу Flood-атак. Комп'ютерно-інтегровані технології: освіта, наука, виробництво. 2015. № 18. С. 6-12.

6. Бевз О.М., Папінов В.М., Скидан Ю.А. Проєктування програмних засобів систем управління. URL: http://posibnyky.vntu.edu.ua/bevz/ (дата звернення 21.01.2024)

7. Воронов М.П. Інформаційне забезпечення діяльності місцевих органів державної влади та органів місцевого самоврядування. Державне управління та місцеве самоврядування. 2011. Вип. 2, ч. 2. С. 106–108.

8. Гаман Т.В. Вдосконалення організаційно-правового механізму інформаційної діяльності місцевих державних адміністрацій : дис. ... канд. наук з держ. упр.: 25.00.02. Л., 2006. 246 с.

9. Гарасимчук О.І., Костів Ю.М. Оцінка ефективності систем захисту інформації. Вісник КНУ імені Михайла Остроградського. 2016. № 1. С. 16–20.

10. Гвозденко М.В., Чобу Я.В. Технічні та програмні засоби виявлення джерела DDoS-атаки. GLOBAL SCIENTIFIC UNITY. 2014. С. 106-115.

11. Геєць В.П., Клебанова Т.С., Іванов В.В. Моделі й методи соціально-економічного прогнозування. Харків: Вид-во ХДЕУ, 2003. 422 с.

12. Гнатюк С. Є. Математичні моделі оцінки та прогнозування надійності програмно-керованих засобів захисту інформації в системі урядового зв'язку. Ukrainian Information Security Research Journal. 2016. № 2. С. 150-156.

13. Гордієнко І.В. Інформаційні системи і технології в менеджменті. 5 вид., перероб. і допов. К.: КНЕУ, 2013. 279 с.

14. Грайворонський М.В., Новиков О.М. Безпека інформаційно-комунікаційних систем / за заг. ред. Академіка НАН України М. З. Згуровського. К. : BHV, 2009. 608 с.

15. Гриценко О. Природа інформаційного суспільства та розвиток світового ринку мас-медіа. Вісник Львівського університету. 2009. № 32. С. 214-222.

16. Грищук Р. В. Атаки на інформацію в інформаційно-комунікаційних системах. Сучасна спеціальна техніка. 2011. № 1(24). С. 61-66.

17. Грищук Р. В. Теоретичні основи моделювання процесів нападу на інформацію методами теорій диференціальних ігор та диференціальних перетворень : моногр.. Житомир : Рута, 2010. 280 с.

18. Діордіца І. Система забезпечення кібербезпеки: сутність та призначення. Інформаційне право. 2017. № 7. С. 109-116.

19. Дубовой В.М., Москвіна С.М., Никитенко О.Д. Моделювання процесів і систем керування. Вінниця, ВНТУ, 2009.103 с.

20. Єрмошин В.В., Хорошко В.О., Капустян М.В. Методика оцінки інформаційних ризиків системи управління інформаційною безпекою. Сучасний захист інформації. 2010. №3. С. 95-104.

21. Журавель Н. О. Організаційна регламентація бізнес-процесів як умова забезпечення їх ринкової безпеки. Управління розвитком. 2014. № 2. С. 121-124.

22. Ілляшенко С. М. Економічний ризик: навч. посіб. К.: Центр навчальної літератури, 2014. 220 с.

23. Квєтний Р. Н., Коцюбинський В. Ю., Кислиця Л. М., Казимірова Н. В., і Кириленко Г. О., Адаптивна система підтримки прийняття рішень з використанням методів нечіткого логічного висновку. НаукПраці ВНТУ, вип. 3, Лис 2011. С. 1-10.

24. Козенков Д.Е. Проєктування бізнес-процесів як основа створення архітектури підприємства. Вісник СумДУ. Серія Економіка. 2011. № 3. с. 126-136.

25. Кравець П., Киркало Р. Системи прийняття рішень з нечіткою логікою. Вісник НУ «Львівська політехніка». 2009. № 650. С. 115-123.

26. Лаврінський Г.В. Моделювання системних характеристик в економіці /Г.В. Лаврінський, О.С. Пшенишнюк, С.В. Устинко, О.Д. Шарапов. К.: ЕКМО, 2004. 169с.

27. Лазарєв Ю. Ф. MATLAB і моделювання динамічних систем. Навчальний посібник. Глава 3. Пакет програм Simulink. Київ: НТУУ «КПІ», 2009. 79 с.

28. Моделювання бізнес-процесів/ уклад. О. І. Подоляка, К. М. Жулінська. Суми : ДВНЗ “УАБС НБУ”, 2013. 20 с.

29. Нейромережева методологія розпізнавання інтернет-орієнтованого шкідливого програмного забезпечення. URL: http://jrnl.nau.edu.ua/index.php/Infosecurity/article/view/4688 (дата звернення 12.01.2024)

30. Нейронні мережі. STATISTICA Neural Networks: Методологія і технології сучасного аналізу даних / за редакцією В. П. Боровикова. 2-е вид. , перероб. і дод. К.: Телеком, 2008. 392 с.

31. Нечітка логіка на прикладах. iSearch. URL: http://www.isearch.kiev.ua/index.php/uk/internetsecurity/837–fuzzy–message (датазвернення: 05.10.2023).

32. Писаревський І. М., Нохріна Л.А., Познякова О.В. Менеджмент організацій: Навчальний посібник. Харків: ХНАМГ, 2018. 133с.

33. Плескач В.Л., Рогушина Ю.В., Кустова Н.П. Інформаційні технології та системи: підруч. для студ. екон. спец. К.: Книга, 2018. 520 с.

34. Пономаренко В. С., Мінухін С. В., Знахур С. В. Теорія та практика моделювання бізнес-процесів: монографія. Харків: Вид. ХНЕУ, 2013. 244 с.

35. Ромашко С.М. Опорний конспект лекцій з дисципліни «Інформаційні системи в менеджменті». Львів: ЛІМ, 2017. 49с.

36. Системи підтримки прийняття рішень : навчальний посіб- ник для самостійного вивчення дисципліни / уклад.: С. М. Братушка, С. М. Новак, С. О. Хайлук. Суми : ДВНЗ «УАБС НБУ», 2010. 265 с.

37. Субботін С. О. Подання й обробка знань у системах штучного інтелекту і підтримки прийняття рішень: Навчальний посібник. Запоріжжя: ЗНТУ, 2008. 341 с.

38. Субботін С.О., Корнієнко О.В. Нейромережеве моделювання залежностей результатів випробувань газотурбінних авіадвигунів. Автоматизація технологічних і бізнес-процесів. 2018. № 10. С. 9-16.

39. Твердохліб М.Г. Інформаційне забезпечення менеджменту : Навч. посібник. К.: КНЕУ, 2012. 224 с.

40. Хмельов О.Г. Моделювання процесів бізнес-прогнозування за допомогою нейромережевих структур. URL: http://www.economy.nayka.com.ua/?op=1&z=38 (дата звернення 08.11.23)

41. Цмоць О.І., Маршук А.А. Прогнозування фінансового стану підприємства за допомогою штучних нейронних мереж. Науковий вісник НЛТУ України, 2011. Вип. 21.9. С.347-352.

42. Яремко С., Кузьміна О., Новицький Р. Використання технологій штучного інтелекту для прогнозування бізнес-процесів. Комп'ютерно-інтегровані технології: освіта, наука, виробництво. 2021. № 43. С. 230-235.

43. Ясинська Н.А., Івченкова О.Ю. Використання нейронних мереж в

моделюванні фінансових результатів бізнес-процесів. Світ фінансів. 2019. № 3(60). С. 108-120.

44. A fuzzy expert system for automatic seismic signal classification. URL: https://www.sciencedirect.com/science/article/pii/S09574114053 (дата звернення: 10.10.2023).

45. A novel fuzzy decision–making system for CPU scheduling algorithm. URL: https://link.springer.com/article/10.1007/s00521–015– 1987–8(дата звернення: 12.10.2023).

46. Abadeh M., Habibi L., Kortos N. Intrusion Detection Using a Fuzzy Genetics–Based Learning. Deli, 2007. P. 314–318.

47. Classification of Network Traffic Using Fuzzy Clustering for Network Security. URL: https://link.springer.com/chapter/10.1007/978– 3–319–62701–4_22 (дата звернення: 12.10.2023).

48. Computer Aided Development of Fuzzy, Neural and Neuro–Fuzzy Systems. URL: https://www.researchgate.net/publication/312590719_Computer_Aided_Development_of_Fuzzy_Neural_and_Neuro–Fuzzy_Systems (дата звернення: 13.10.2023).

49. Diagnosing computer hardware failures using expert system (rule–based technique). URL: https://www.researchgate.net/publication/79205502_DIAGNOSING_COMPUTER_HARDWARE_FAILURES_USING_EXPERT_SYSTEM_RULEBASED_TECHNIQUE (дата звернення: 01.10.2023).

50. Expert diagnosis of computer systems, neuro–fuzzy knowledge base. IEEE: web–site. URL: https://ieeexplore.ieee.org/document/metrics#metrics(дата звернення: 29.09.2023).

51. Expert evaluation model of the computer system diagnostic features. URL: https://ieeexplore.ieee.org/document/7027101/metrics(дата звернення: 29.09.2023).

52. Mamdani, E.H., Assillan, S.: An experiment in linguistic synthesis with afuzzy logic controller.Int. J. Man-Mach. Stud. 7(1), 1–13, 1975.

53. Network–based output tracking control for T–S fuzzy systems using an event–triggered communication scheme. URL: https://www.sciencedirect.com/science/article/pii/S0165011032(дата звернення: 21.09.2023).

54. Simulation and Model–Based Design. Mathworks: web–site. URL: https://www.mathworks.com/simulink.html(дата звернення: 16.10.2023).

55. Stuart Russell and Peter Norvig Artificial Intelligence: A Modern Approach: Fourth edition (2020). Hoboken: Pearson. https://lccn.loc.gov/2019047498.

56. The neuro–fuzzy diagnostic model synthesis with hashed transformation in the sequence and parallel mode. URL: https://ric.zntu.edu.ua/article/view/101022/96247(дата звернення: 02.10.2023).

57. Usability Determination Using Multistage Fuzzy System. Sciencedirect: web–site. URL: https://www.sciencedirect.com/science/article/pii/S1842 (дата звернення: 08.10.2023).

58. Yu.Yu. Gromov, O.G. Ivanova, V.V. Alekseev and ets. Intelligent information systems and technologies: textbook. Tambov: FGBOU VPO «TSTU», 2013. 244 p.

ДОДАТОК А

```
[System]
Name='fuzzysys'
Type='mamdani'
Version=2.0
NumInputs=3
NumOutputs=1
NumRules=13
AndMethod='min'
OrMethod='max'
ImpMethod='min'
AggMethod='max'
DefuzzMethod='centroid'

[Input1]
Name='Temperature'
Range=[0 1]
NumMFs=3
MF1='mf1':'gbellmf',[0.2083 2.5 -0.00529100529100525]
MF2='mf2':'gbellmf',[0.2083 2.5 0.5]
MF3='mf3':'gbellmf',[0.2083 2.5 1]

[Input2]
Name='Speed'
Range=[0 100]
NumMFs=3
MF1='mf1':'gbellmf',[20.84 2.5 -1.332e-15]
MF2='mf2':'gbellmf',[20.83 2.5 50]
MF3='mf3':'gbellmf',[20.84 2.5 100]

[Input3]
Name='Congression'
Range=[0 0.5]
NumMFs=3
MF1='mf1':'gbellmf',[0.1042 2.5 5.205e-18]
MF2='mf2':'gbellmf',[0.1042 2.5 0.25]
MF3='mf3':'gbellmf',[0.1042 2.5 0.5]

[Output1]
Name='system'
Range=[0 10]
NumMFs=3
MF1='mf1':'trimf',[-4.167 0 4.167]
MF2='mf2':'trimf',[0.8333 5 9.167]
MF3='mf3':'trimf',[5.833 10 14.17]

[Rules]
1 1 1, 1 (1) : 1
2 2 2, 2 (1) : 1
3 3 3, 3 (1) : 1
1 2 3, 2 (1) : 1
1 3 2, 2 (1) : 1
2 1 3, 2 (1) : 1
3 1 2, 2 (1) : 1
3 2 1, 2 (1) : 1
2 3 1, 2 (1) : 1
3 2 1, 2 (1) : 1
3 2 2, 3 (1) : 1
2 3 3, 3 (1) : 1
3 2 3, 3 (1) : 1
```