Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

АТЕСТАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Нейромережеві методи побудови математичних моделей нелінійних об'єктів

(тема)

Виконав:
студент II курсу, групи КСМм-19-1
Тищенко О.І.
(прізвище, ініціали)

Спеціальність
123 – Комп'ютерна інженерія
(код і повна назва спеціальності)

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

Освітня програма
Комп'ютерні системи та мережі
(повна назва освітньої програми)

Керівник: проф. Руденко О.Г.
(посада, прізвище, ініціали)

Допускається до захисту

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

2020 р.

Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

Спеціальність 123 – Комп'ютерна інженерія
(код і повна назва)

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

Освітня програма Комп'ютерні системи та мережі
(повна назва)

ЗАТВЕРДЖУЮ:
Зав.кафедри _____
(підпис)
"____" _____ 20__р.

ЗАВДАННЯ

НА АТЕСТАЦІЙНУ РОБОТУ

студентові Тищенку Олегу Ігоровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Нейромережеві методи побудови математичних моделей нелінійних об'єктів

затверджена наказом по університету від " 30 " жовтня 2020 р. № 1487Ст

2. Термін подання студентом роботи до екзаменаційної комісії 14 грудня 2020 р.

3. Вхідні дані до роботи

1) документація по нейромережевим моделям аналізу
2) документація мови програмування Java
3) перелік використаних програмних засобів: ОС Windows 10
4) інтегроване середовище: Java і Aglets Software Development Kit (ASDK)
5) особливості нейромережевого аналізу

4. Перелік питань, що потрібно опрацювати в роботі

1) дослідити існуючі методи та варіанти побудови моделей поведінки об'єктів
2) розробити генеративну модель поведінки
3) розробити комплексну модель об'єктів системи, яка буде враховувати динамічні та статичні властивості поведінки, а також їх зміни
4) провести верифікацію програми(тестування)
5) впровадити агентську технологію та об'єктно-орієнтований підхід у систему моніторингу діяльності користувачів комп'ютерних систем в корпоративних мережах

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Демонстраційні матеріали. Плакати – 12 арк. ф. А4

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

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз сучасного стану проблеми та методів	03.11.2020-6.11.2020	
	її вирішення		
2	Дослідження існуючих рішень	7.11.2020-10.11.2020	
3	Формування переліку вимог до програми	11.11.2020	
4	Розробка та тестування програми	12.11.2020-28.11.2020	
5	Оформлення пояснювальної записки	29.11.2020-11.12.2020	

Дата видачі завдання 2 листопада 2020 р.

Студент _____
(підпис)

Керівник роботи _____ проф. Руденко О.Г.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка атестаційної роботи: 92 с., 24 рис., 1 табл., 2 дод., 20 джерел.

НЕЙРОМЕРЕЖА, БАГАТОАГЕНТНА СИСТЕМА, МОДЕЛЬ, ТРЕНД, АНАЛІЗ, КОРИСТУВАЧ СИСТЕМИ, ПОВЕДІНКОВА МОДЕЛЬ, СЕАНСОВА МОДЕЛЬ, ГЕНЕРАТИВНА МОДЕЛЬ.

Метою атестаційної роботи є розробка програми для аналізу поведінки об'єктів систем та застосування в системах моніторингу і виявленні некоректних дії, їх усунення.

У ході виконання атестаційної роботи досліджено існуючі рішення, їх переваги, недоліки та сформовано функціонал. Розроблена комплексна нейромережева модель поведінки користувачів комп'ютерної системи, що створена на нейромережевому підході і включає три складові: інтерактивну складову, яка оцінює динаміку поведінки об’єктів, сеансову складову, яка забезпечує статистичні властивості поведінки об’єктів та модуль аналізу, призначений для виявлення можливих змін в поведінці об’єктів.

Для вирішення поставленого завдання використовується платформа Aglets Software Development Kit разом із мовою програмування JAVA.

ABSTRACT

Master’s thesis: 92 pages, 24 figures, 1 tables, 2 appendices, 20 sources.

NEURAL NETWORK, MULTI-AGENT SYSTEM, MODEL, TREND, ANALYSIS, SYSTEM USER, BEHAVIORAL MODEL, SESSION MODEL, GENERATIVE MODEL.

The purpose of the certification work is to develop a program for analyzing the behavior of system objects and its application in monitoring systems and identifying incorrect actions, their elimination.

In the course of attestation work the existing solutions, their advantages, disadvantages and functionality are investigated. A comprehensive neural network model of computer system user behavior has been developed, which is based on the neural network approach and includes three components: an interactive component that evaluates the dynamics of object behavior, a session component that provides statistical properties of object behavior and an analysis module possible changes in the behavior of objects.

The Aglets Software Development Kit together with the JAVA programming language is used to solve this problem.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

АБС – автоматизована банківська система

АРМ – автоматизоване робоче місце

ІАЦ – інформаційно-аналітичний центр

СУБД – система управління базами даних

DAL – рівень доступу до даних (англ., Data Access Layer)

EF – сутність організації (англ., Entity Framework)

ERP – планування ресурсів (англ., Enterprise Resource Planning)

GM – генеративна модель (англ., Generative models)

MS – мультиагентна система (англ., MultiAgent System)

MVC – модель контролер (англ., Model-View-Controller)

NM – нейронна модель (англ., Nequral models)

ORM – зображення відношень об’єктів (англ., Object Relation Mapping)

RAD – швидка розробка додатка (англ., Rapid Application Development)

SQL – структурована мова запитів (англ., Structured Query Language)

TA – аналіз трендів (англ., Trend Analysis)

ВСТУП

На сьогодні масштабне використання комп'ютерних технологій майже у всіх сферах людської діяльності призвело до більшої уваги до самого користувача. Адже інформація про те, які дії він виконує (або мусить виконувати), можна застосовувати в різних сферах. Наприклад, в системах спостереження (стеження) за персоналом (Personal Security Programs), системах безпеки, при створенні екосистеми для користувача, що персоналізуються, в Web-додатках і т. д. Тому розробка методів аналізу і моделей поведінки об’єктів комп'ютерних систем є актуальною темою [4-6].

Проте існуючі методи і підходи до аналізу поведінки користувачів комп'ютерних систем недостатньо враховують всі аспекти його роботи, що зумовлює складність людино-машинної взаємодії. Тому, в деяких системах враховуються тільки динамічні властивості поведінки об’єктів, але ніяк не аналізується статистична інформація про їх діяльність. І навпаки, існують системи, в яких модель поведінки об’єктів базується на використанні статистичних даних, але при цьому аналіз його дії в режимі реального часу не проводиться [1]. При цьому в більшості систем ніяк не враховуються можливі тренди зміни поведінки об’єктів системи.

Саме тому, поведінка об’єктів комп'ютерних систем є складний процес, в даний час є актуальною розробка ефективних моделей комплексного аналізу поведінки користувачів. Вирішенню саме цієї задачі присвячена дана атестаційна робота.

Метою роботи є розробка якісних моделей аналізу поведінки об’єктів комп'ютерних систем і їх застосування в системах моніторингу і виявлення незвичної діяльності. Для досягнення поставленої мети в атестаційній роботі вирішаються наступні питання, а саме:

- дослідження та класифікація існуючих методів та підходів до побудови поведінкової моделі об’єктів комп'ютерних систем;

- створення комплексної моделі користувачів комп'ютерних систем, яка враховуватиме як динамічні, так і статистичні властивості їх поведінки, а також можливі зміни в їх поведінці;

- розробка генеративної моделі поведінки користувачів; забезпечення верифікації розробленої моделі на реальних даних;

- розробка та реалізація системи моніторингу діяльності користувачів комп'ютерних систем в корпоративних мережах з використанням об'єктно орієнтованого підходу і агентної технології [8].

В атестаційній роботі використано методи системного аналізу складних множинних процесів, нейромережевий підхід, об'єктно-орієнтований підхід до моделювання, статистичне моделювання та розробка складних систем; агентська парадигма з методом чисельних експериментів.

Розроблена комплексна нейромережева модель поведінки користувачів комп'ютерних систем, яка базується на нейромережевому підході та складається з трьох компонентів: інтерактивний (прогнозний) компонент, що враховує динаміку поведінки користувачів, сеансовий (статистичний) компонент, що враховує статистичні властивості поведінки користувача і аналізатор трендів, призначений для виявлення можливих змін в поведінці об’єктів [7]. На відміну від існуючих підходів, запропонована модель дозволяє забезпечити комплексний підхід до аналізу поведінки користувачів як під час його роботи (у режимі реального часу), так і після закінчення сеансу (у відкладеному режимі). В атестаційній роботі розроблено генеративну модель поведінки користувачів, яка забезпечує репрезентативні масиви даних для верифікації та ідентифікації комплексної моделі.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз методів і підходів до аналізу поведінки об'єктів

В даній ситуації для аналізу поведінки об’єктів використано багато різних методів та підходів. Проте у більшості, вони всі базуються на аналізі та виявленні закономірностей або часто повторюваних дій користувачів з цілю вирішення задач автоматизації, прогнозування їх поведінки, виявлення дивностей в їх роботі та автоматична адаптація комп'ютерної системи до потреб юзерів (користувачів). Інформація, яка використовуються для побудови моделей, враховує базові знання про індивідуальні характеристики об’єктів, які визначають його поведінку. Для обробки цих даних використовуються як прості методи, засновані на аналізі часто повторюваних дій (history-matching methods), так і складні методи машинного навчання [2].

Через те, що для побудови поведінкової моделі об’єктів найчастіше виникають складні задачі прогнозування та класифікації, було розглянуто різні підходи, які застосовуються для вирішення цих питань.

1.1.1 Методи класифікації та прогнозування

Згідно досліджень зарубіжних та вітчизняних вчених кількість всіляких методів та прийомів, методик прогнозування перевищило сотню, але число базових методів, що повторюються в різних інтерпретаціях в інших методах, не перевищує десятка. Під методом прогнозування розуміється множина прийомів прогнозування, що пропонують на основі ретроспективних даних, відомих зовнішніх і внутрішніх зв'язків підсистем, а також їх вимірювань, вивести думки певної достовірності щодо майбутнього системи. Розрізняють декілька базових методів прогнозування:

- логічні і структурні методи штучного інтелекту;

- математичні методи тимчасової екстраполяції;
- математичні методи просторової екстраполяції;
- методи моделювання процесів розвитку;
- евристичні методи прогнозування.

Логічні та структурні методи прогнозування розглядаються тільки в рамках однойменних методів розпізнавання образів, де під образами маються на увазі прогнозовані явища і процеси. Виняток становить ряд методів прогнозування, найбільш поширений з яких заснований на морфологічному аналізі. Морфологічний аналіз пов'язаний з аналізом структурних взаємозв'язків між об'єктами, явищами та концепціями. Логічні методи прогнозування використовують логічні методи розпізнавання прогнозованих явищ та створені на дискретному аналізі і численні висловів. Структурні методи прогнозування використовують спеціальні граматики, що породжують мови, які складаються з пропозицій, кожне з яких описує те, що належить до конкретного класу прогнозних станів. Недоліки цих методів характерні для методів штучного інтелекту: труднощі відпрацювання мови опису прогнозованих класів, витягування інформації з бази даних, а також складність прогнозованої моделі [11].

Залежно від використаного математичного апарату та спрямованості, математичні методи миттєвої екстраполяції можна умовно розділити на три частини:

- методи імовірнісного прогнозування;
- методи аналітичного прогнозування;
- методи статистичної класифікації.

Потреба у імовірнісному прогнозуванні багатовимірних процесів визначається сильним впливом багатьох факторів, які мають випадковий характер. Переважають випадкові складові при вимірюванні, які приводять до великих випадкових змін функцій стану [13, 14]. Для дослідження безперервного прогнозу використовуються оптимальні фільтри. Наприклад, розглянемо фільтр Вінера – Хопфа призначеного для прогнозування

стаціонарних процесів та фільтр Кальмана для нестаціонарних процесів. До недоліків групи імовірнісних методів прогнозування багатовимірних процесів відносяться: наявність представницького об'єму статистичних даних про процеси заміни параметрів; неможливість обліку стрибків у просторі прогнозування; неможливість провести дослідження без математичного опису процесів заміни параметрів.

До методів аналітичного прогнозування багатовимірних процесів відноситься градієнтний метод, цілю якого є функція стану, яка екстраполюється у напрямі вектора градієнта функції стану. Існує ряд методів аналітичного прогнозування, котрі включають похідні змін функції стану. До таких методів включаються: операторний метод та метод підсумовування похідних. До загальних недоліків методів аналітичного прогнозування необхідно вказати великий об'єм обчислень при визначенні прогнозних значень параметрів, а також погрішність у результатах прогнозування при некоректно вибраній моделі.

У теорії статистичної класифікації (розпізнавання образів) процес встановлення екстраполяційних зв'язків здійснюється на основі абстрактної інформації. Недоліком цього підходу є обов'язкова наявність абстракцій. Саме тому, необхідно зробити вибірку даних по об'єкту одного типу з об'єктом, стан якої необхідно спрогнозувати.

Прогнозування параметрів стану у вигляді тимчасової екстраполяції характеристик задіяна в якості аргументу параметр часу. Просторова екстраполяція пов'язана з прогнозуванням в множині характеристик, і полягає в оцінюванні значень векторного поля за детальним спостереженням. Для вирішення цих завдань використовуються різні методи апроксимації та метод багатовимірної лінійної екстраполяції, але якщо кількість вимірювань невелика [5].

Залежно від використовуваного математичного апарату і цільової спрямованості, математичні методи тимчасової екстраполяції умовно поділені на три групи, а саме: методи аналітичного прогнозування, методи

імовірнісного прогнозування та методи статистичної класифікації. До методів аналітичного прогнозування процесів відноситься градієнтний метод, в основі якого функція стану екстраполюється у напрямі вектора градієнта функції стану. Також існує ряд методів аналітичного прогнозування, які враховують похідні змін функції стану. До цих методів відносять операторний метод та метод підсумовування похідних. А до загальних недоліків методів аналітичного прогнозування можна віднести великий об'єм обчислень при визначенні прогнозних значень параметрів і неточність результатів прогнозування при неправильно вибраній моделі.

Необхідність імовірнісного прогнозування багатовимірних процесів визначається сильним впливом безлічі чинників, що мають випадковий характер походження. Переважають випадкові складові при вимірюваннях, які призводять до великих змін функцій стану. А для отримання безперервного прогнозу використовуються оптимальні фільтри названі раніше: фільтр Вінера-Хопфа (для прогнозування стаціонарних процесів), фільтр Кальмана (для нестаціонарних процесів). До загальних недоліків більшості імовірнісних методів прогнозування багатовимірних процесів можна віднести: наявність представницького об'єму статистичних даних про процеси зміни параметрів, неможливість обліку стрибків на ділянці прогнозування та неможливість обійтися без математичного опису процесів зміни параметрів [3].

У теорії статистичної класифікації механізми встановлення екстраполяційних зв'язків здійснюються на основі базової інформації. Недоліком цього підходу є обов'язкова наявність абстрактної інформації. Отже, це необхідна вибірка даних по об'єкту одного типу з об'єктом, стан якої необхідно прогнозувати.

Прогнозування параметрів стану у вигляді тимчасової екстраполяції постановок використовує тільки параметр часу. Просторова екстраполяція пов'язана з дослідженням в просторі характеристик, і полягає в оцінці значень векторного поля за окремими спостереженнями. Для вирішення

цього завдання використовують різні методи апроксимації та методи багатовимірної лінійної екстраполяції, але при невеликій кількості вимірювань.

Як правило, виділяють три базові методи математичних моделей: фізичний, математичний та імітаційний. Фізичний метод дозволяє відтворювати функціонування зокрема елементів і підсистем об'єкту із збереженням його фізичної складової. В свою чергу математичний метод припускає опис за допомогою сукупності співвідношень чи нерівностей, логічних умов та прогнозованих характеристиках стану. Недоліки математичного методу є типові для методів тимчасової екстраполяції. Але через складність об'єкту математична модель функціонування може бути дуже незручною для її оперативного використання. Імітаційний метод застосовується у тому випадку, коли досліджувані процеси складні і багатогранні, тому математична модель стає дуже чітким наближенням до реальних прикладів.

Метод евристичне прогнозування полягає в інтуїтивному виборі з незліченної множини обставин тільки важливі рішення. Більша частина цих домислів полягає в напівсвідомому порівнянні системою всіх величин і варіантів, за допомогою яких швидко усувається все некоректне та непотрібне. Але при цьому евристичні методи суб'єктивні і діють тільки тоді, коли існують системи, які проводять прогнози ситуацій [7].

Розглянемо алгоритм класифікації K-найближчих сусідів. В основі даного методу полягає правило ближнього елемента, що є одним з простих прикладів навчання на основі пам'яті та відповідно до цього правила весь минулий досвід накопичується у великому сховищі правильно класифікованих прикладів вигляду FI-FO:

$$\{(x , d)\}N ,$$

де x_i – вхідний вектор;

d_i – відповідний йому бажаний вихідний сигнал.

Тоді за наявності тестового прикладу в його множину додається найближчий до нього приклад. Наприклад, вектор $\mathbf{x}_N\{\mathbf{x}_1,\mathbf{x}_2,...,\mathbf{x}_N\}$ вважається найближчим сусідом вектора *x-test*, якщо виконується умова:

$$\min d(xi, xtest) = d(xN, xtest),$$

де $d(x_i, x_{test})$ – евклідова відстань між векторами x_i і x_{test}.

Клас, до якого відноситься найближчий сусід, вважається також класом тестового вектора x_{test}. Це правило не залежить від розподілу, який використовується при створенні прикладів навчання. Проводилося формальне дослідження правила найближчого сусіда, що використовується для вирішення задачі класифікації образів. При цьому аналіз ґрунтується на таких припущеннях:

- приклади (x_i, d_i), що класифікуються, незалежні і рівномірно розподілені відповідно до сумісного розподілу прикладу (x, d);
- розмірність навчальної множини N нескінченно велика.

Доведено, що при двох виразах ймовірність помилки класифікації при використанні правила найближчого елемента втричі перевищує ймовірність Байєса (ймовірність помилок). Байєсівська ймовірність помилки – це мінімальне значення помилок на множині всіх правил ухвалення рішення. У цьому контексті вважається, що половина класифікаційної інформації для навчальної множини нескінченного розміру міститься в даних про найближчий елемент [9].

Варіацією класифікатора на основі найближчого елемента є класифікатор К-найближайших сусідів, його можна описати наступним чином:

- знаходимо k класифікованих елементів, які знаходяться біля вектора x_{test}, де k –ціле число;
- вектор x_{test} відносимо до класу, котрий частіше за інших

зустрічається серед *k* найближчих елементів тестового вектора.

Отже, класифікатор на основі k найближчих елементів працює подібно до пристрою усереднювання. До його недоліку можна віднести такі причини: метод не враховує одиничне спостереження, а метод потенційних функцій базується на гіпотезі про складові функцій, що розділяють множини, які відповідають різним класам. Значною перевагою цього методу є зведення багатьох відомих алгоритмів розпізнавання. У разі використання аналізу дискримінанта розмежувальні поверхні визначаються скалярними функціями:

$$g1(k), ..., gm(k),$$

де $k=\{k_1, ..., k_n\}$ – вектор стану системи.

Функції g(k), названі дискримінантами, і вибираються так, щоб для всіх k (де R клас) виконувалася умова g (k)>gr(k), r, =1, ..., u, r. За допомогою функцій дискримінантів можна отримати стандартний і зручний метод завдання розділення поверхонь, але для чіткого визначення функцій необхідно мати повну інформацію про об’єкти, що підлягають класифікації, але часто це буває неможливо.

1.1.2 Методи аналізу на основі марківських ланцюгів

Багато прогнозних моделей об’єктів системи (описують послідовності дій, що виконуються об’єктом, з цілю прогнозу) базується на застосуванні марківських ланцюгів. Це пов'язано з такими висновками про характер поведінки об’єктів: ймовірність виконання наступної дії залежить від декількох попередніх. Про такі процеси говорять, що вони мають миттєву пам'ять [16]. Це наштовхує на думку моделювати такі послідовності даних за допомогою марківських ланцюгів і прихованих марківських моделей (HMM – Hidden Markov Model). Проте, не зважаючи на здатність статичних моделей

описувати об’ємний клас послідовностей, існують ефективні процедури їх налаштування, які мають багато негативних моментів. Справа в тому, що розмір марківських ланцюгів експоненціальний та збільшується із зростанням порядку, тобто кількість станів відповідного ЦА поводиться як O(L), де O – розмір алфавіту символів, а L – порядок ланцюга. Отже, практичну цінність мають тільки моделі з дуже невеликим порядком.

Проте в цьому випадку вони неякісно апроксимують послідовності дій, що виконуються об’єктами. Відповідно до прихованих марківських моделей, то окрім теоретично доведеної складності їх навчання, спроби їх практичної реалізації показали необхідність дуже великих витрат для налаштування [15].

Наведемо приклад систем, які використовують даний підхід для побудови моделей об’єктів.

У системі розглядається побудова змішаної (глобальної) моделі Web-користувачів та створення на цій базі персоніфікованої імовірнісної моделі для кожного з них. Мета такого підходу це прогнозування дій користувача на основі попередніх дій. Для цього використовуються метод ентропії та марківські ланцюги. Сама ідея запропонованого методу полягає в наступному: спочатку будується глобальна модель поведінки на основі аналізу дій тих об’єктів, для яких є дані (деяка абстрактна інформація). Кожному з них присвоюється індивідуальних ваговий коефіцієнт в змішаній моделі. Потім будь-який інший об’єкт, що створював Web-вузол, відноситься до тієї або іншої групи об’єктів. Тобто поведінка нового об’єкта або об’єкт з незначним набором даних (у статистичному сенсі) може бути наближена до вже відомої і, таким чином бути прогнозованою.

У системі пропонується модель, яка використана для прогнозування команд, що виконуються об’єктом в операційній системі Linux, а також допомагає користувачеві донабирати команди: тобто на основі k введених символів вибирати найбільш ймовірну команду. Для вирішення поставленого завдання авторами розроблений спеціальний метод поетапного імовірнісного моделювання дій, який заснований на використанні марківських ланцюгів. У

цій системі взято за основу звичайний марківський ланцюг першого порядку, тобто він не враховує інформацію більшої довжини, ніж один запит (команда). Науковий дослід з даними по 46 об’єктах показав 42.6% чіткість досліджень.

1.1.3 Методи на основі аналізу дій, які часто повторюються

Даний підхід аналізу заснований на побудові шаблонів часто повторюваних дій, які виконують об’єкти системи, наприклад, в операційній системі, Web-браузері або інших програмних продуктах. Під час роботи об’єкта його поточна поведінка порівнюється з побудованою моделлю шаблонів. При цьому результати порівняння можуть використовуватися для автоматичного виконання набору дій, а також для порівняння дій користувача. Основною перевагою даного підходу до побудови моделей об’єктів є його простота, але до його недоліків слід віднести наступні фактори, а саме: складна побудова моделей, що персоналізуються, для кожного об’єкта, в них ніяк не враховуються взаємозв'язки між побудованими шаблонами [16].

В якості прикладу систем, які використовують даний підхід для побудови моделей об’єктів, розглянемо систему в якій застосовується адаптивний помічник, а саме операційна система Linux. У системі він використовується для вибору найбільш ймовірних URL-адресів в Web-браузерах, причому система пропонує до того ж враховувати їх пріоритет. У системі цей підхід використовується для створення прогнозуючого призначеного для користувача інтерфейсу (predictive user interface).

1.1.4 Методи аналізу на основі нейронних мереж

У багатьох системах для аналізу поведінки об’єктів комп'ютерних систем застосовуються нейроні мережі, що забезпечує інтелектуальний і

адаптивний підхід до аналізу та узагальнення даних. Ці властивості дозволяють нейронним мережам вирішувати складні завдання (у контексті побудови моделей об’єктів) прогнозування, класифікації, фільтрації і т. д. Розглянемо деякі приклади використання нейромережевого підходу до створення моделей поведінки користувачів. У мережі запропонована система NNID (Neural Network Intrusion Detector), яка ідентифікує користувачів на основі обмеженої кількості команд (100 операцій), введених за добу та однієї нейронної мережі, яка побудована для всіх об’єктів системи [17]. При цьому враховується тільки кількість викликання кожної команди, а не їх послідовність. Ці кількості певним чином кодуються і є варіацією нейронної мережі. Якщо сеанси користувачів не потрапляють під їх нормальний шаблон, тобто користувач не ідентифікований або ідентифікований неправильно, то генерується відповідне повідомлення про помилку. Були отримані відмінні результати по комп'ютерній системі, що складається із 7 користувачів. Очевидно, що це достатньо приємні умови, оскільки в реальних системах кількість користувачів може досягати декількох сотень і більшість виконуватиме однотипні запити. Тому побудова однієї нейромережевої моделі для ідентифікації всіх користувачів, яка базується тільки на кількості команд, може бути неефективною.

У іншій системі ідентифікація користувача проводилася на основі послідовності команд. Нейронна мережа використовувалася для прогнозування заданих команд. При цьому команди кодувалися таким чином: спочатку вони були пронумеровані в довільному порядку, а потім кожній команді ставився у відповідність вектор – двійковий запис його номера. Проте реальний вихід нейронної мережі не був двійковим: компоненти вихідного вектора складали реальні числа. Тому в даному випадку критерій відмінності був не наочним. Також варто відзначити, що модель була побудована тільки для роботи з аудитом – файлами Linux-систем.

1.2 Області застосування поведінкових методів аналізу

1.2.1 Системи безпеки та стеження

Сьогодні є популярними так звані системи стеження (Security Programs), які в основному використовуються комерційними компаніями для спостереження за діями персоналу. Застосування таких систем дозволяє виявити зловмисників при просочуванні інформації, або ж визначити, чи використовують співробітники робочі станції в своїх особистих цілях. Такі програми, як PC Spy, Inlook Express, Paparazzi дозволяють знімати та зберігати зображення (копії екранів), на яких можна бачити, які програми запускаються користувачами: браузери із завантаженими Web-сторінками, вікна з електронними повідомленнями. Тобто в даному випадку в якості профілю об’єкта дослідження використовується графічна інформація про запущені віконні додатки. Проте такі системи володіють рядом недоліків, серед яких слід виділити великий об'єм даних, що зберігаються, і необхідність мануального налаштування частоти “знімання” копій екрану. Тобто, якщо частота збереження даних невелика, то виявити атаки буде складно. Інакше, в системі доведеться зберігати велику кількість зображень (копій екрану), що приведе до збільшення навантаження на комп'ютерну мережу, що є поганим для системи [11].

Відомо, що достатньо велика кількість атак на комп'ютерну систему здійснюється із середини. Деякі джерела вказують на те, що близько 75-85% атак ініціюється саме об’єктами всередині комп'ютерної системи. Це може відбуватися внаслідок здійснення підміни акаунтів, перевищення користувацьких прав, використання недоліків в підсистемі захисту або ж внаслідок використання помилок в ПЗ (програмному забезпеченні). На жаль, традиційні методи захисту, такі як аутентифікація, обмеження прав доступу виявляються неефективними при виявленні помилок в роботі користувачів. Тому одним з сучасних методів вирішення цієї проблеми є використання

систем виявлення і запобігання вторгненням IDS Intrusion detection systems і IPS Intrusion prevention systems. У загальному випадку системи IDS і IPS поділяються на такі, що використовують моделі нападів, і такі, які засновані на аналізі поведінки.

При використанні методів виявлення вторгнень, заснованих на сигнатурах атак, застосовуються знання, раніше накопичені про атаки і вразливості комп'ютерної системи. Система виявлення вторгнення містить інформацію про ці вразливості і виявляє спроби їх використання. Якщо така спроба виявлена, система генерує повідомлення про тривогу. Іншими словами, будь-яка дія, яка явним чином не визначено як атака, розглядається прийнятною. Перевагами такого підходу є теоретично низький рівень помилкових тривог. Недоліками є складність збору інформації про відомі напади, необхідність постійного оновлення баз даних уразливостей та сигнатур, практична неможливість виявлення атак, що ініціюються об’єктами всередині комп’ютерних систем.

У свою чергу, методи виявлення і запобігання вторгненням на основі поведінки припускають, що вторгнення може бути виявлене на основі відхилення від стандартної або очікуваної поведінки системи або об’єктів. Модель стандартної поведінки будується на основі інформації, зібраної впродовж деякого часу роботи системи або роботи об’єктів. Система виявлення вторгнень порівнює цю модель з поточною інформацією. Якщо відхилення виявлене, здійснюється сповіщення про тривогу. У системах запобігання вторгнень, до того ж, можуть здійснюватися автоматичні дії із запобігання атакам. Іншими словами, все, що не відповідає раніше аналізованій поведінці, розглядається як вторгнення.

Переваги методу виявлення атак на основі аналізу поведінки полягають в тому, що вони можуть виявити спроби використання нової або непередбаченої вразливості, а також виявити незвичайну роботу внутрішніх об’єктів комп'ютерної системи. Високий рівень помилкових тривог вважається головним недоліком цих методів. Це пов'язано з тим, що

неможливо налаштувати впродовж фази навчання всі можливі варіанти поведінки комп'ютерної системи або його об’єктів. Крім того, поведінка може змінюватися з часом, що приводить до потреби періодичного перенавчання поведінки системи.

При побудові систем виявлення аномалій використовуються наступні підходи: статистичні моделі, виявлення аномалій з використанням експертних оцінок, моделі на основі марківських ланцюгів, нейромережеві моделі. Так, в системі виявлення атак виконується на основі однієї моделі, яка будується для всіх користувачів комп'ютерної системи. В якості профілю об’єктів використовується інформація про кількість введених команд за сеанс. При цьому для всіх об’єктів використовується один і той же набір із 500 команд [5]. Для ідентифікації користувачів застосовується апарат нейронних мереж. Так, якщо сеанс користувача не потрапляє під нормальний шаблон, тобто користувач не ідентифікований або ідентифікований неправильно, то генерується відповідне повідомлення про напад на систему. Серед недоліків пропонованого підходу можна виділити наступні:

- необхідність перенавчання моделі при додаванні нових користувачів;
- використання обмеженого набору команд може привести до того, що профілі користувачів відрізнятимуться трохи.

У свою чергу, в роботі ідентифікація об’єктів здійснюється на основі послідовності команд. Для кожного об’єкта будується нейронна мережа, яка навчається прогнозувати наступну команду даного об’єкта на основі декількох попередніх. Якщо команди, отримані в результаті нейромережевого прогнозу, істотно відрізняються від реально введених користувачем.

1.2.2 Системи навчання

Моделі поведінки об’єктів активно використовуються в системах віддаленого навчання. Це обумовлено необхідністю звикання системи та

учбового процесу до потреб тих суб’єктів, які навчаються .

Наприклад, в системі з метою ефективного обміну інформацією використовується модель викладача та студента. Модель студента базується на інформації про його взаємодію з системою у минулому (звіт про вивчений матеріал, помилки при виконанні різних завдань) і є комбінацією стереотипних моделей, що створені (overlay). Стереотипна модель використовується з цілю класифікації знань та навиків студента на різні рівні: новачок, початківець, експерт. На сам перед, модель, яка створюється, є масивом пари «поняття/значення» області дослідження. Модель студента використовується для дослідження тенденцій в здійсненні помилок, що повторюються, для формування звіту про успішність кожного студента, для перевірки та наскільки він засвоює матеріал. Модель викладача використовується для забезпечення допомоги при організації навчального процесу [13].

У даній роботі розглядається завдання прогнозування поведінки студентів в навчальних системах. Для цього використовується модель dual, що складається з двох частин: ранньої моделі (fresh model) та розширеної моделі (extended model). Для побудови ранньої моделі використовуються останні дані про роботу учня. Але розширена модель враховує всю історію поведінки студента в системі навчання. Прогнозування дій учня здійснюється на основі обох моделей.

1.2.3 Web-системи

Моделі поведінки суб’єктів часто застосовуються у Web-системах, зокрема для створення адаптивних Web-вузлів та додатків.

Так, в системі методологія, заснована на аналізі повторюваних дій, використовується для вибору URL-адресів в Web-браузерах. В першу чергу, в системі пропонується вдосконалений метод, який враховує спочатку пріоритет URL-адресів. Пріоритет обчислюється як зважена сума наступних

характеристик: кількість запитів до Web-сторінки, відносна частота переглядів та кількість переглядів в послідовностях. До недоліку запропонованого підходу можна віднести те, що в ньому ніяк не враховані взаємозв'язки між Web- сторінками, які відвідує суб’єкт. Тобто ніяк не враховується контент, який переглядав користувач.

У системі пропонується використовувати спеціалізований додаток WebWatcher, який супроводжує користувача по кожній сторінці під час перегляду Web-ресурсів та прогнозує посилання, які вибере користувач. Для цього використовуються знання про область інтересів конкретного суб’єкта дослідження, а також докладна інформація про вміст Web-сторінок та взаємозв'язки між ними. Результати експериментів встановили точність 37,8%. Варто зазначити, що запропонований підхід враховує тільки поточні інтереси суб’єкта, але ніяк не враховує історію його поведінки.

У системі пропонується підхід, що дозволяє стежити за діями об’єкта під час перегляду Web-стрінок. На основі отриманої інформації про його роботу будується відповідна блок-схема (моделлю користувача). Для витягування корисної інформації з блок-схеми використовується спеціально розроблена мова MINT, яка схожа на SQL.

У системі пропонується розробка адаптивного Web-вузла, який миттєво формує Web-сторінки релевантні інтересам користувача. Для цього використовується концептуальний метод добування знань, який аналізує аудит - файли Web-сервера.

У даній роботі розглядається побудова змішаної (глобальною) моделі Web-користувачів і створення згодом персоніфікованої імовірнісної моделі для кожного з них. Мета даного підходу прогнозування дій користувача (таких як перегляд документів або пошук) на основі попередніх. До недоліків даного підходу можна віднести використання витратних статистичних методів, необхідність перенавчання моделі при додаванні нових об’єктів системи, а також практична неможливість застосування в режимі реального часу. При цьому для збору даних про роботу об’єктів системи автори

використовують механізм cookies (збереження даних в браузері), який в загальному випадку може бути відключений користувачем системи. Слід також відзначити точність при визначенні періоду роботи користувача, якщо відвідувач Web-вузла не виконує дії впродовж 200 секунд, вважається початок нового.

Моделі об’єктів також застосовуються для розробки систем адаптивного інформаційного пошуку, насамперед в мережі Internet. Наприклад, в системі пошук інформації здійснюється на основі примітивної контекстної моделі об’єкта. Дана модель є багатошаровою семантичною мережею, кожен шар якої відповідає деякому контексту, а її вузли є ключовими словами. Запропонована система реалізована за допомогою агентського підходу. У даній роботі описується мульти - агентська система пошуку на основі моделей об’єктів [15]. Запропонована система містить такі типи агентів, які взаємодіють між собою: для фільтрації інформації (агент відповідальний за побудову і підтримку профілю користувача) та виявлення нових властивостей в поведінці користувачів (адаптація профілю об’єкта до різних інформаційних ресурсів). При цьому агенти розвиваються шляхом застосування генетичних алгоритмів, такі як: мутації, клонування, кроссовер.

1.2.4 Клієнтоорієнтована система

Моделі поведінки користувачів широко застосовуються при створенні так званих призначених для користувача оболонок (екосистем), що персоналізуються. Дані системи використовуються з метою автоматичної адаптації середовища до поведінки об’єкта, який дозволяє збільшити ефективність його взаємодії з різними додатками.

Так, в системах пропонується адаптивний помічник командного рядка (command-line), який прогнозує дії, які виконуються користувачем в операційній системі Linux, а також дозволяє йому автоматично заповнювати команди. Для цього використовується спеціально розроблений алгоритм

IPAM (Incremental Probabilistic Action), точність прогнозу якого досягає 47,3%. До недоліку цього методу можна віднести те, що в ньому ніяк не враховуються налаштування команд, що є істотним зауваженням у взаємодії об’єктів під управлінням даної ОС (операційна система) [18].

Також у системі пропонуються спеціально розроблені методи машинного навчання (machine learning techniques), які здійснюють прогнозування команд. Отримані відмінні результати (точність прогнозу в деяких випадках досягає 74-89%), але на невеликому об'ємі даних реальних об’єктів системи. Проте така висока точність досягнута за рахунок використання потужної надбудови над ОС, що дозволяє збирати різноманітну інформацію про ввід/вивід даних, який здійснюється запущеними командами, в першу чергу це зчитування файлів.

Також тут використано найпростіші моделі, в яких профіль користувачів враховує тільки часто повторювані набори. Саме тому цей підхід застосовується з метою створення адаптивного помічника, який виводить перелік найбільш ймовірних дій, які будуть виконані об’єктом в командному рядку операційної системи Linux.

1.2 Постановка задачі

У атестаційній роботі встановлено завдання розробки комплексного підходу до аналізу поведінки комп’ютерної мережі з метою виявлення нестандартних запитів в їх роботі. Даний підхід повинен враховувати динамічні і статистичні властивості поведінки, а також можливі зміни в поведінці, не пов'язані з аномаліями у діях.

На сьогодні актуальна задача розробки моделей поведінки користувачів комп'ютерних систем, які враховуватимуть як динамічні, так і статичні властивості поведінки користувачів, а також можливі тренди їх поведінки. Під динамічними властивостями розумітимемо послідовність дій.

Аналіз існуючих моделей поведінки об’єктів, проведені в попередньому розділі, довели, що в більшості випадків вони є простими і не дозволяють описати всі аспекти поведінки. Так, в багатьох моделях виявлення нестандартних дій об’єктів здійснюється в рамках одного операційного циклу. Наприклад, існують моделі, які враховують тільки динамічні властивості поведінки об’єктів, але при цьому не аналізують його статистичні властивості. Тобто моніторинг і виявлення дивностей відбувається в режимі реального часу під час сеансу роботи об’єктів. І навпаки, існують моделі, які засновані тільки на аналізі статистичної інформації про діяльність об’єктів за сеанс в цілому [3]. У таких моделях виявлення аномальної діяльності здійснюється у відкладеному режимі. У більшості робіт також не ставиться питання про виявлення змін поведінки об’єктів, не пов'язаних з аномалією, і визначення моменту часу, коли модель необхідно редагувати (зміна поведінки об’єктів).

Статистичними властивостями об’єктів називаються інтегральні характеристики роботи користувача в перебігу сеансу.

При побудові моделей поведінки об’єктів можна виділити наступні загальні етапи:

- збір і попередня обробка даних про роботу користувачів;
- аналіз даних з метою виділення інформативних ознак або зменшення розмірності даних;
- розробка методів обробки даних і побудова моделі;
- верифікація моделі і інтерпретація отриманих результатів, у атестаційній роботі будуть розглянуті всі перераховані етапи.

Після того, як для кожного об’єкта комп'ютерної системи побудована модель, виявлення нестандартної діяльності повинне відбуватися наступним чином, а саме результати поточної роботи користувача відповідають раніше побудованій моделі, отже таку поведінку можна вважати нормальною, в іншому випадку аномальною.

Тут було розглянуто області застосування моделей поведінки юзерів

комп'ютерних систем, проаналізовані деякі підходи до їх побудови та визначені їх переваги і недоліки.

Аналіз показав, що моделі поведінки об’єктів широко застосовуються для вирішення задач в різних областях, наприклад, в системах безпеки та стеження, у Web-додатках, в системах навчання та для виявлення шахрайства. При цьому для аналізу дій користувача застосовуються різноманітні методи і підходи. Практично всі вони засновані на виявленні закономірностей в його роботі шляхом прогнозування або класифікації (нормальна/аномальна поведінка). У даному розділі розглянуті існуючі методи прогнозування і класифікації, зокрема методи на основі побудови шаблонів дій користувачів, які часто повторюються (наприклад, в операційній системі, Web - браузері або інших програмах), а також методи, засновані на застосуванні марківських ланцюгів і нейронних мереж. Аналіз існуючих методів прогнозування і класифікації показує, що найбільш ефективним при вирішенні складних нестаціонарних задач є підхід на основі нейронних мереж.

Існуючі моделі в більшості випадків є простими і нажаль, не дозволяють розібрати всі аспекти поведінки об’єктів КС. Тому, розроблені моделі враховують тільки динамічні властивості поведінки користувачів, але при цьому не аналізують його статистичні властивості [7, 8]. І навпаки, існують моделі, які засновані на аналізі статистичної інформації про діяльність об’єктів та при цьому не дозволяють аналізувати поведінку під час сеансу його роботи(режим реального часу).

Отже, актуальною задачею у дослідженні нейронних мереж є розробка ефективних методів системного аналізу та моделей поведінки об’єктів КС, які враховуватимуть як динамічні, так і статичні (інтегральні) властивості поведінки користувачів, а також можливі тренди його поведінки. У даній роботі пропонується комплексний підхід до аналізу поведінки об’єктів КС з метою виявлення аномалій в його роботі, який дозволяє врахувати аспекти.

2 АНАЛІЗ ВИКОРИСТОВУВАНИХ ТЕХНОЛОГІЙ

2.1 Комплексна модель діяльності об'єктів КС

У даному розділі обґрунтовується комплексний підхід до аналізу поведінки користувачів комп'ютерних систем з метою моніторингу і виявлення аномальної діяльності. Пропонується загальна структура комплексної моделі і опис її складових (інтерактивної, сеансової і модуля аналізу трендів), аналізуються переваги запропонованого підходу в порівнянні з існуючими моделями і підходами.

2.1.1 Структура комплексної поведінки об'єктів

Для цілісного опису поведінки користувачів комп'ютерних систем і виявлення недоліків у їх роботі пропонується використовувати комплексний підхід, який включає в себе три модулі:

- інтерактивний модуль, що описує роботу об’єкта під час сеансу на рівні виконуваних команд (запущених процесів);
- сеансовий модуль, що засновано на аналізі інтегральних даних про роботу об’єкта за сеанс в цілому;
- модуль аналізу трендів, що дозволяє виявити можливі тренди в роботі об’єкта.

Інтерактивний та сеансів модулі комплексної схеми пропонується будувати на основі нейронних мереж. Застосування нейронних мереж забезпечує інтелектуальний підхід до аналізу і узагальнення даних, отриманих під час роботи об’єкта [14]. У загальному випадку існують різні нейромережеві парадигми. Наприклад, асоціативні мережі, які використовуються для вирішення задач асоціативної пам'яті і відновлення зашумленних образів, мережі Кохонена, які використовуються для

кластеризації образів, мережі на основі радіальних базисних функцій (РБФ), призначені для вирішення задач класифікації образів.

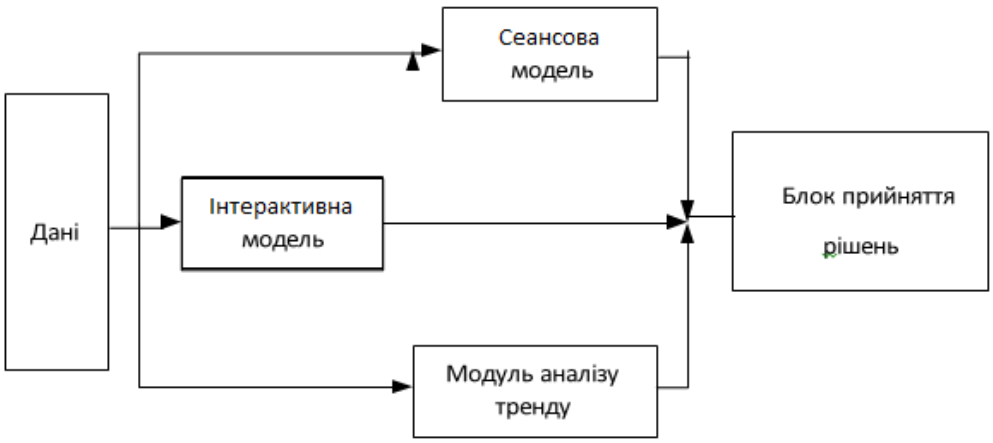

Рисунок 2.1 – Структура комплексної моделі поведінки об’єктів

Для вирішення задач даної роботи вірним рішенням є багатошарова мережа прямого розповсюдження інформації. Використання мереж такого типу обумовлене тим, що згідно теореми Колмогорова вони є універсальними апроксимаціями та можуть ефективно застосовуватися як для вирішення задач прогнозування, так і класифікації. Застосування нейронних мереж одного і того ж виду в різних складових комплексної моделі також забезпечує переваги при створенні моделі. Реалізувавши одну нейромережеву парадигму, її можна використовувати як в інтерактивній, так і в сеансовій складовій, що дозволяє забезпечити мультисерверність пропонованого підходу.

Конкретний вид комплексної моделі об’єкта обумовлює використання різних типів нейронних мереж прямого розповсюдження. У разі інтерактивної складової нейронна мережа прогнозує команди об’єкта на основі попередніх. Відповідно вона використовується як інтерполятор [4]. У разі сеансової складової, прослідковується, що нейронна мережа використовується як класифікатор і на основі інтегральних даних, які були

отримані за сеанс, визначає, на скільки активність користувача відповідає раніше побудованій моделі.

2.1.2 Нейронні мережі поширення інформації

Штучні нейронні мережі є простою моделлю нервових тканин живих організмів, клітини якої є аналогією до базових компонентів сучасних нейрон – модулів. Свій потенціал нейронні мережі беруть з паралельної обробки інформації та із властивості узагальнення. Під терміном узагальнення розуміється властивість отримувати обґрунтований результат, які не зустрічалися в процесі навчання мережі [10]. Ці властивості дозволяють нейронній мережі вирішувати складні задачі прогнозування, класифікації, обробки зображень та управління. При цьому використання нейронних мереж забезпечує наступні корисні властивості систем, а саме:

- не лінійність;
- відображення вхідної інформації у вихідну;
- адаптивність;
- відмово стійка;
- масштабованість.

Найбільш універсальним видом нейронних мереж є мережі прямого розповсюдження. Нейронна мережа прямого розповсюдження це багатошарова структура взаємозв'язаних простих оброблювальних елементів. Зв'язки між елементами одного шару і зворотні зв'язки в мережі відсутні.

Для нейронних мереж прямого розповсюдження інформація від виходу шару n до виходу шару n+1 розповсюджується таким чином:

$$y_i^{n-1} = f(s_i), y^{n-1} = f(s^{n-1}),$$

- де n+1 – це номер шару;
- j – це індекс нейрона в шарі n+1(j=1, Nn 1);

- Nn 1 – це число нейронів в шарі n+1;
- f – це нелінійна активаційна функція шару n+1;
- s – це постсинаптичний потенціал j-го нейрона.
-

$$s_i^{n+1} = \sum_{k=1}^{N} W_{jk} y_k^n + b_i^{n-1} S^{n-1} = W^{n+1} y^{\sim n}$$

- де W – ваговий коефіцієнт зв'язку k-го нейрона шару n з j-м нейроном шару n+1;
- y – вихід k-го нейрона шару n;
- y – розширений вектор з врахуванням Bias-нейрона;
- b – поріг j-го нейрона шару n+1;
- W – матриця d(d-1).

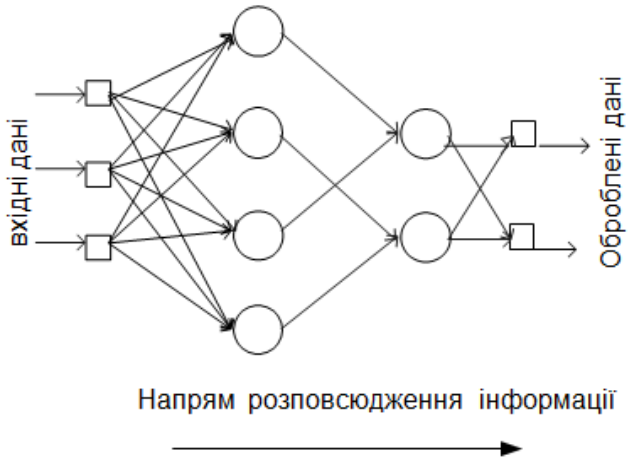

Рисунок 2.2 – Приклад архітектури штучної нейронної мережі

В якості активаційної функції використовується сигмоїдна функція, що має такий вигляд:

$$f(x) = \frac{1}{1 + e^{-ax}} \big(f(x) \in (0,1)\big).$$

Її вибір обґрунтований тим, що вона є нелінійним порогом та безперервною, що диференціюється [17]. При цьому похідна визначається через самостійну функцію:

$$f(x) = \alpha f(x[1 - f(x)].$$

Параметр α визначає переваги даної функції. Для нейронної мережі можна виділити наступні режими функціонування:

- навчання вільних коефіцієнтів мережі;
- тестування роботи мережі на вибірці даних;
- використання мережі на невідомих даних.

2.1.3 Інтерактивна модель поведінки

В результаті аналізу файлів аудиту для різних об’єктів виявлено, що кожен об’єкт мережі володіє характерною поведінкою, у яку закладено виконання характерних послідовностей команд в різних сеансах. Для різних об’єктів така поведінка різна, але для кожного об’єкта можна виявити закономірності, характерні тільки для нього. Саме на цьому і засновано використання інтерактивної моделі. Вона забезпечує прогнозування дій об’єктів на основі зроблених команд, тим самим імітуючи динамічні властивості поведінки об’єкта. Для цього використовується нейромережевий підхід. Це зумовлено тим, що поведінкою об’єктів є складний нелінійний процес, а також необхідністю прогнозування в режимі реального часу і виявлення прихованих закономірностей в його роботі [19]. У основу інтерактивної моделі поведінки об’єктів покладена нейронна мережа прямого розповсюдження.

При реалізації інтерактивної моделі для кожного об’єкта будується нейронна мережа, яка навчається так, щоб при подачі на вхід мережі послідовності з декількох команд на виході отримувати наступну команду. В даному випадку нейронна мережа використовується в режимі інтерполятора.

Нехай s(t->{1,2….} – номер сеансу), тобто це сеанс роботи об’єкта системи, для якого є наступна послідовність виконуваних команд:

$$c = c_1, c_2, \dots, c_n,$$

- де N – це кількість команд введених за сеанс st;
- c->A – це десятковий номер i-ої введеної команди сеансу st;
- A – це алфавіт команд.

Перш ніж подати послідовність із m команд на вхід нейромережі, використовується бінарний код. Це зроблено з метою збільшення інформаційної ємкості вхідних образів. Для кожної команди c, її десятковий номер перетворюється на двійковий код, що складається з q біт. Тобто, кожній команді ставиться у відповідність бінарний вектор c->0,1(k=1,m) Після цього послідовність із m команд встановлюється шляхом об’єднання побудованих двійкових векторів с кожної команди: x=c1, c2, c3…. Таким чином, результат роботи нейронної мережі при виконанні i-1команди визначається рівнянням:

$$c = F(x)x = c_1, c_2, \dots, c_m,$$

- де F – це нелінійне перетворення, яке здійснюється нейронною мережею;
- xc – це вхід та вихід мережі вектора;
- m – це глибина пам'яті.

На основі кількості команд, які були правильно обчислені нейронною мережею, створено висновок про те, чи відповідає поточна поведінка об’єкта раніше побудованої моделі [9]. Нижче наведено структуру цієї моделі поведінки.

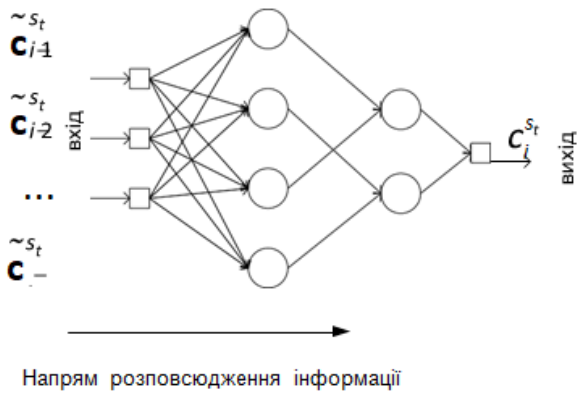

Рисунок 2.3 – Структура інтерактивної моделі

Величина (i)->[0,1] визначає відносне число вірних команд прогнозованих нейронною мережею до моменту введення і команд. Тобто значення (i) менше деякого порогу, то поведінку об’єкта слід вважати нетиповою, інакше нормальною.

При побудові інтерактивної моделі необхідно враховувати, що об’єктам системи властиво змінювати поведінку з часом. Тому з метою забезпечення адаптації до їх поведінки нейронну мережу слід періодично навчати, тренувати нові властивості. Саме для цього і призначений модуль аналізу трендів, який буде описаний нижче.

Слід зазначити наступні переваги нейромережевої моделі об’єктів в комп'ютерних системах на основі послідовності команд:

- адаптація до зміни поведінки об’єктів;
- незалежність від кількості об’єктів в системі, оскільки кожен цикл навчання розглядається окремо;
- можливість виявлення прихованих закономірностей в поведінці об’єктів системи завдяки використанню інформації не тільки про те, які команди були введені, але і про послідовності цих команд [8].

2.1.4 Сеансова модель та модуль аналізу

На відміну від інтерактивної моделі, сеансова ґрунтується на використанні інтегральних (статистичних) даних, отриманих під час роботи учня за сеанс в цілому. При цьому враховується наступна інформація:

< n, o, h, d, s >

- де n – кількість команд, які були виконані користувачем протягом сеансу;
- o – результати інтерактивної моделі, тобто відносна кількість правильно прогнозованих команд за сеанс;
- h – номер комп'ютера в мережі, за яким працював користувач;
- d – тривалість сеансу;
- s – час початку сеансу.

Цей набір даних використовується в якості вхідних даних для виявлення нормальної або дивної роботи об’єкта мережі у сеансі. Як і у разі інтерактивної моделі, для вирішення задачі класифікації використовується нейронна мережа прямого розповсюдження [5]. Тобто для кожного об’єкта системи будується нейронна мережа, яка навчається так, щоб на основі статистичної інформації за сеанс відносити поведінку об’єкта до класу нормального або нестандартного. При цьому очікуваний вихід нейронної мережі може приймати два значення: y для нормальної поведінки користувача і n для дивного, тобто нейронна мережа працює як класифікатор.

Тепер, як і раніше s(t->{1,2….} – це сеанс роботи студента, після закінчення якого є наступний набір даних n, o, h, d, s. Тоді вихід нейронної мережі після закінчення сеансу s визначається наступним співвідношенням:

$$\Delta = F(x), x = (n, o, h, d, s),$$

- де F – нелінійне перетворення, яке здійснюється мережею;
- $x\Delta$ – вхід і вихід мережі;
- параметри n, o, h, d, s визначені в співвідношенні.

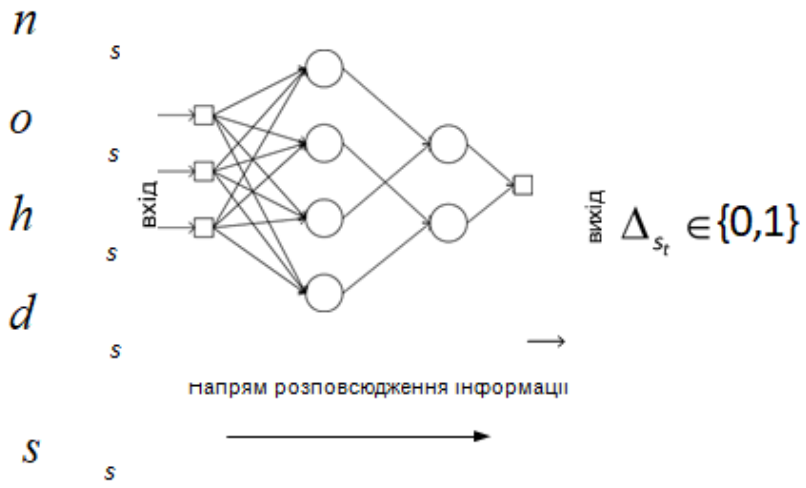

Рисунок 2.4 – Структура сеансової моделі

При реалізації сеансової моделі застосовувалося таке кодування вхідної інформації:

- для кількості команд використовувалося десяткове кодування;
- тривалість сеансу користувача вимірювалася в хвилинах і нормувалася на 8 годин (тривалість робочого дня);
- результати інтерактивної моделі представлялися числом в проміжку [0, 1], що визначає процентне співвідношення правильно прогнозованих команд;
- набір хостів кодувався цілими числами 1, 2, 3 ...;
- час початку сеансу представлявся цілими числами від 0 до 23 і нормувалося на 24.

У роботі показано, що якщо при навчанні нейронної мережі бажаний вихід приймає два значення (наприклад, 0 і 1; тобто нейронна мережа розділяє вхідний простір на два класи), отже подача на її вхід незалежно

образу на виході буде отримана ймовірність приналежності цього образу до одного або іншого класу. Таким чином, значення D належатиме відрізку [0;1] і визначатиме ймовірність нормальної (відповідної моделі) поведінки користувача.

Слід виділити наступні переваги сеансової моделі об’єкта:

- незалежність від кількості об’єктів в системі, оскільки для кожного блоку побудована своя нейромережева модель;
- сеансова модель об’єкта враховує статистичні параметри, що дозволяє виявляти дивності, не виявлені інтерактивною моделлю об’єкта;
- адаптація до зміни поведінки об’єктів.

З практики відомо, що поведінка системи не є стаціонарним процесом і з часом може змінюватися. Це може відбуватися з різних причин, насамперед через використання нових версій ПЗ. В цьому випадку, навчені на застарілих даних інтерактивна та сеансові моделі адекватно не описуватимуть поведінку користувача, а поведінка, що змінилася, інтерпретуватиметься як нестандартна. Відповідно, ефективність використання цих моделей знизиться. Тому в комплексну модель поведінки об’єктів пропонується додати модуль аналізу трендів, що забезпечує виявлення «трендів» поведінки і що надає додаткове підтвердження дивної поведінки об’єктів системи. Проте при цьому необхідно враховувати, що дивну поведінку також можна інтерпретувати як зміна поведінки [3]. Тому при побудові модуля аналізу трендів враховуємо наступне:

- вважається, що дивна поведінка об’єктів характеризується різкими змінами і швидкоплинністю;
- у свою чергу, природна зміна поведінки відбувається впродовж декількох сеансів і не характеризується різкими перепадами.

Нехай A – алфавіт команд деякого об’єкта системи. Передбачається, що за цей проміжок часу поведінка користувача не змінювалася. Позначимо за допомогою N – кількість команд в алфавіті. При цьому кожній команді поставимо у відповідність номер від 1 до N. Номер N+1 алфавіту

зарезервуємо за всіма тими командами, які не були виконані під час сеансів. Тому (t>T) – поточний сеанс роботи об’єкта, для якого є наступна послідовність виконаних ним команд:

$$c=c_1, c_2, \ldots., c_n,$$

тоді $c_i = N + 1$, якщо $c_i\{1, \ldots., N\}$.

Для того, щоб визначити, чи змінилася поведінка користувача, після закінчення сеансу st будується вектор g(st), компоненти якого визначаються таким чином:

$$g(s(t)) = \begin{cases} 1, \text{якщо існує таке } k = 1, N, \text{що } c = j. \\ 0, \text{в іншому випадку} \end{cases}$$

Тобто якщо певна команда була виконана під час сеансу st, то значення відповідної компоненти вектора g(st) стає рівним 1, інакше 0. Приведемо приклади векторів g(st), побудованих для різних сеансів одного і того ж об’єкта дослідження:

g(s_1) = (1 1 0 1 1 0 1 1 1 0 1 1 1 0 0 0 0 0 0 1 0 1 …)

g(s_2) = (1 0 0 1 0 0 1 1 1 0 1 1 1 0 0 0 1 0 0 1 0 1 …)

g(s_3) = (1 0 1 1 0 0 1 1 1 0 1 1 1 0 1 1 0 0 0 1 0 1 …)

Потім цей вектор g(st) попарно порівнюється з аналогічними векторами, побудованими для попередніх сеансів st-1, st-2, …, st-l (у даній роботі бралося l = 5). Як міра порівняння використовується відстань Хеммінга:

$$K(g(s),g(s(t))=\sum x(g(s),g(s)),$$

де x(g(s),g(s)) $+\begin{cases} 1, \text{якщо } g(s) \neq g(s) \\ 0, \text{в іншому випадку} \end{cases}$.

Тобто величина визначає число компонентів двох векторів, значення яких різні [15]. Отримавши в результаті порівняння l значень, обчислюємо середнє і нормуємо його на загальну кількість команд в алфавіті:

$$H = \frac{1}{N}\left(\sum k\left(g(s), g(s)\right)\right).$$

Якщо поведінка користувача не змінилася і не дивна, тоді вектор g(st) буде відрізнятиметься від попередніх. Відповідно, значення параметра Ht буде невеликим (меншим деякого порогу H*: Ht < H*). Та навпаки, якщо спостерігається дивна поведінка об’єкта, вектор g(st) значно відрізнятиметься від всіх останніх і значення параметра Ht буде великим (більшим деякого порогу H*: Ht < H*). Якщо ж Ht (H*; H*), можна говорити про природну зміну поведінки об’єктів.

Отже, критерій зміни поведінки об’єкта системи можна сформулювати в наступному вигляді, а саме: для даного сеансу об’єкта st на основі формул обчислюється значення параметра Ht. Якщо Ht < H*, вважається, що поведінка не змінилася, якщо H* < Ht < H* – змінилася, якщо ж Ht > H* – тоді поведінка нестандартна.

2.2 Асоціативна комплексна модель

Виходячи з приведеного опису модулів комплексної моделі, її загальна структура може бути уточнена.

Таку структуру прийнято називати асоціативною машиною (committee machine). Асоціативні машини зазвичай застосовуються для вирішення складних задач шляхом їх розбиття на множину невеликих і простих задач з подальшим об'єднанням отриманих рішень. При навчанні обчислювальна

простота досягається за рахунок розподілу задачі навчання серед множини так званих експертів. Асоціативна машина інтегрує знання, накопичені спеціалістами, в загальне рішення, яке має пріоритет над кожним рішенням окремого спеціаліста.

Залежно від підходів для об'єднання рішень окремих експертів в загальне розрізняють дві основні категорії асоціативних машин: статичні (static structure) і динамічні структури (dynamic structure).

У статичних структурах рішення різних експертів об'єднуються за допомогою деякого механізму, який не враховує вхідний сигнал. Ця категорія структур працює на основі наступних методів [6].

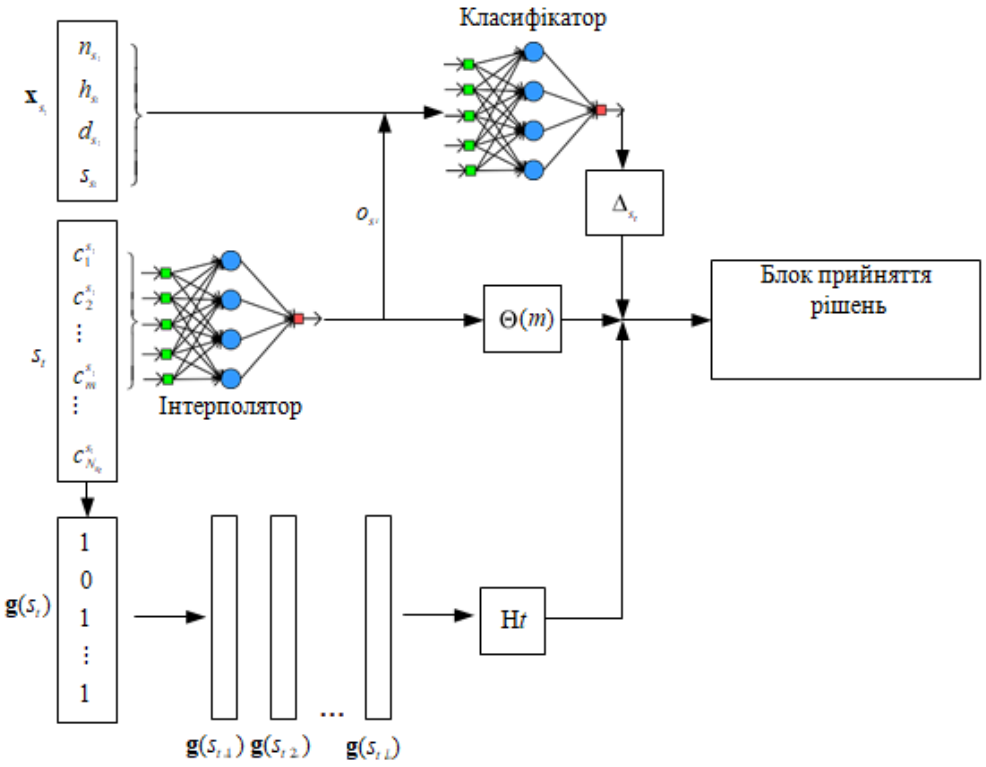

Рисунок 2.5 – Структура комплексної моделі

Усереднювання по ансамблю (ensemble averaging), при якому вихідний сигнал обчислюється як лінійна комбінація виходів окремих експертів.

Посилення (boosting), при якому експерти навчаються на різних підмножинах даних.

У свою чергу, в динамічних структурах (dynamic structure) асоціативних машин вхідний сигнал безпосередньо враховується в механізмі об'єднання вихідних сигналів експертів. Можна виділити дві різні реалізації динамічних структур.

Змішування думок спеціалістів (mixture of experts), при якому думки окремих спеціалістів об'єднуються в єдину шлюзову мережу (gating network) та ієрархічне об'єднання думок спеціалістів, при якому відгуки окремих спеціалістів об'єднуються за допомогою декількох шлюзових мереж, організованих в ієрархічну структуру.

Асоціативна машина, яка використовується в даній атестаційній роботі є динамічною структурою, що використовує методологію змішування думок спеціалістів. В ролі спеціалістів в даному випадку виступають нейромережеві моделі інтерактивної і сеансової складових і модуль аналізу трендів. Аналізуючи різні властивості поведінки користувача, кожен зі спеціалістів (нейронні мережі і модуль аналізу трендів) дає своє рішення про нормальну або нестандартну поведінку об’єкта. За допомогою блоку ухвалення рішення асоціативна машина, в свою чергою, ухвалює загальне рішення.

Так, під час сеансу роботи користувача рішення про характер поведінки користувача ухвалюється на основі інтерактивної моделі об’єкта: якщо (i)<(де величина(i) обчислюється згідно формули), то поведінка користувача вважається дивною, інакше нормальною.

Після закінчення сеансу роботи користувача, окрім кількості вірно спрогнозованих команд за сеанс (N) , є також результати роботи сеансової моделі і модуля моніторингу змін Ht [6]. Відповідно, загальне рішення про характер поведінки об’єктів ухвалюється з урахуванням всіх трьох величин. У даній моделі використовується лінійна комбінація величин:

$$\Omega = \alpha(N) + \beta\Delta + \gamma(1 - H),$$

де α, β, γ – коефіцієнти нормувань $\alpha + \beta + \gamma = 1, \alpha, \beta, \gamma \in [0,1]$.

Таким чином, виходячи з визначення величин можна скласти наступне: чим менше значення, тим більша ймовірність того, що поведінка користувача за сеанс була нестандартною.

Коефіцієнтам α, β, γ можна також дати імовірнісну інтерпретацію: ці значення є ймовірністю правильної класифікації поведінки користувача за сеанс відповідної складової комплексної моделі. Тому вибір значень α, β, γ можна здійснити виходячи з таких міркувань: на основі тестових даних про роботу користувача за сеанс (які включають як приклади аномальної поведінки, так і нормальної) обчислюють відповідні частоти правильної класифікації його поведінки кожної складової комплексної моделі. Тоді коефіцієнти α, β, γ слід вибирати пропорційними цим частотам. У цій атестаційній роботі всі три моделі однаково правильно відносили поведінки об’єктів до класу нормального чи дивного, тому всі коефіцієнти α, β, γ вибрано рівними 1/3. Проте питання вибору коефіцієнтів заслуговує уважного розгляду в рамках окремого дослідження, яке виходить за межі поставлених в даній роботі задач.

2.3 Генеративна модель поведінки об'єктів КС

2.3.1 Побудова генеративної моделі

У загалі функціонування нейронної мережі значно залежить від якості навчальної вибірки. Справа в тому, що при невеликому розмірі навчальної множини нейронна мережа має тенденцію до жорсткого запам'ятовування образів, що приводить до зменшення її властивості до узагальнення. Так, при побудові інтерактивної складової в даній роботі проблема з навчальною вибіркою даних не виникала, оскільки навіть за нетривалий період часу

роботи об’єкта мережі може бути зібрана достатня кількість образів для навчання нейронної мережі. Наприклад, з урахуванням того, що об’єкт в середньому вводить від 97 до 124 команд за сеанс, то за десять сеансів навчальна вибірка може налічувати до 2000 образів.

У разі сеансової складової нейромережевою моделлю є класифікатор. Тому для стійкого розділення класів нормальної та нестандартної поведінки потрібно забезпечити навчальну множину достатньо великого розміру. Проте для сеансової складової складно забезпечити навчальну вибірку даних, оскільки вхідними даними в даному випадку служить узагальнена інформація про сеанс роботи об’єкта. Тобто, на відміну від інтерактивної складової, в сеансовій кожен сеанс роботи об’єкта визначає всього один навчальний образ (або точку в просторі ознак). Так, за три місяці число таких сеансів може досягати 100, що в нашому випадку недостатньо для якісного навчання нейронної мережі і оптимізації її архітектури. Для вирішення цієї проблеми можна використовувати два підходи. Перший з них полягає в значному збільшенні відрізку часу, в рамках якого відбувається спостереження за поведінкою об’єкта на довгий час. Проте в цьому випадку велика ймовірність того, що за цей проміжок часу вона зміниться і, таким чином, навчальна множина міститиме суперечливі образи. Другий підхід полягає в статистичному моделюванні даних на основі наявної вибірки (побудова генеративної моделі поведінки об’єкта) [7].

Оскільки в сеансовому модулі для навчання нейронної мережі використовується інформація про кількість команд, що вводяться, за сеанс, номері комп'ютера, тривалості і часі початку сеансу, для кожного об’єкта необхідно промоделювати саме цей набір даних.

Нехай є дані про роботу користувача за декілька місяців. Задача полягає в тому, щоб на основі цих даних висунути гіпотезу про їх імовірніс-

ні характеристики, а потім перевірити ступінь відповідності цієї гіпотези реальним даним. В якості критерію згоди використовуватимемо так званий критерій Х.

Розглянемо принципи побудови критерію X та отримані залежності детальніше.

Нехай для кожного сеансу st є наступний набір даних про роботу користувача: n,h,d,s. Кожну з цих величин можна розглядати як випадкову, яка приймає T певних значень. До того ж, вважатимемо ці величини незалежними. Позначимо за допомогою X одну з цих випадкових величин. Розіб'ємо її значення по k інтервалах і представимо у вигляді наступного статистичного ряду.

Таблиця 2.1 – Статичний ряд

I_i	X_1,X_2	X_2,X_3	…	$X_k; X_{k+1}$
I^{n*}	1^{n*}	N^{2*}	…	N_k^*

Потрібно перевірити, чи узгоджуються ці дані з гіпотезою про те, що випадкова величина X має закон розподілу, заданий функцією розподілу F(x) або щільністю f(x) (назвемо її теоретичною).

Знаючи теоретичний закон розподілу, можна знайти теоретичну імовірність попадання випадкової величини в той або інший інтервал:

$$p_1, p_2, \ldots, p_k.$$

Перевіряючи узгодженість теоретичного і статистичного (емпіричного) розподілів, виникає питання про те, яким же способом слід вибирати міру розбіжностей. В якості такої міри в математичній статистиці зазвичай вибирають суму квадратів -pi pi , взятих з деякими вагами ci:

$$U = \sum_{i=1}^{k} c_i p_i - p_i.$$

Коефіцієнти c вводяться тому, що відхилення, що відносяться до різних інтервалів, не можна вважати рівноправними по значущості. Дійсно, одне і те ж по абсолютній величині відхилення p(p) може бути малозначним, якщо сама ймовірність p велика, і дуже помітна, якщо вона мала. Тому природно вибирати ваги c обернено пропорційними ймовірності p [7]. Саме том, вченими було доведено, що якщо покласти c то при достатньо великих значеннях n закон розподілу величини U практично не залежатиме від функції розподілу F(x) і числа n, а залежатиме тільки від кількості інтервалів k і із збільшенням n наближатися до розподілу χ зі щільністю:

$$f(u) = \frac{1}{2^2\Gamma\left(\frac{r}{2}\right)} e^2 (u > 0),$$

- де $\Gamma(\alpha) = \int_0^\infty t^{\alpha-1} edt$ – гамма функція;
- r – кількість мір свободи.

Таким чином, міра розбіжності матиме такий вигляд:

$$\chi^2 = U = \sum n \frac{p_i - p_i^2}{p_i}$$

Оскільки розподіл χ^2 залежить від параметра r (число мір свободи), то для його обчислення використовують наступне правило: $r=k-s$.

До прикладів таких умов можна віднести вимогу, щоб сума всіх частот була рівна одиниці, або вимога рівності теоретичного середнього і дисперсії статичному і т. д.

Для розподілу χ^2 складені спеціальні таблиці. Користуючись ними, можна для кожного значення χ^2 та числа мір свободи знайти ймовірність того, що величина, розподілена за законом χ^2, перевершить це значення.

Таким чином, схема застосування критерію χ^2 до оцінки узгодженості статистичного і теоретичного розподілів зводиться до наступного:

- визначається міра розбіжності χ^2 відповідно до формули;
- визначається число мір свободи r як різниця між числом інтервалів k і кількістю накладених зв'язків s;
- по набутих значеннях χ^2 і r визначається ймовірність p.

Якщо ця ймовірність вельми мала, гіпотеза про відповідність статистичного розподілу теоретичному відкидається. Якщо ж ця ймовірність відносно велика, дану гіпотезу можна визнати такою, що не суперечить досвідченим даним.

2.3.2 Закон розподілу

Доведено, що за допомогою критерію вдалося визначити теоретичні розподіли для параметрів сеансової моделі. Тепер на їх основі необхідно згенерувати відповідні набори значень для заданої кількості сеансів. Ця задача зводиться до задачі генерування значень випадкової величини, розподіленої по заданому закону. Скористаємося для вирішення задачі наступним твердженням, що дозволяє генерувати значення [13].

Нехай ξ – випадкова величина з безперервною функцією розподілу F_ξ. Тоді випадкова величина $\eta=F_\xi(\xi)$ буде рівномірно розподілена на відрізку [0; 1]. Таким чином, необхідно спочатку згенерувати випадкову величину η, розподілену рівномірно на відрізку [0; 1], а потім підставити набутих значень у функцію, зворотну до функції розподілу F, тобто $F^{-1}(\eta)$.

2.3.3 Апроксимація залежностей

Розглянемо отримані залежності при моделюванні. В якості теоретичних розподілів використовуються рівномірні, нормальні та

логарифмічні. Для різних об’єктів був проведений аналіз різних розподілів та виявлено значення критерію χ^2 в середньому рівне 3,4, було отримано для логарифмічного розподілу. Оскільки в даному випадку кількість вимірів свободи рівна 3 (k = 7, s = 4), це значення забезпечило 72% відповідності гіпотези реальним даним. Нижче наведено теоретичний та емпіричний розподіл параметра для декількох об’єктів системи.

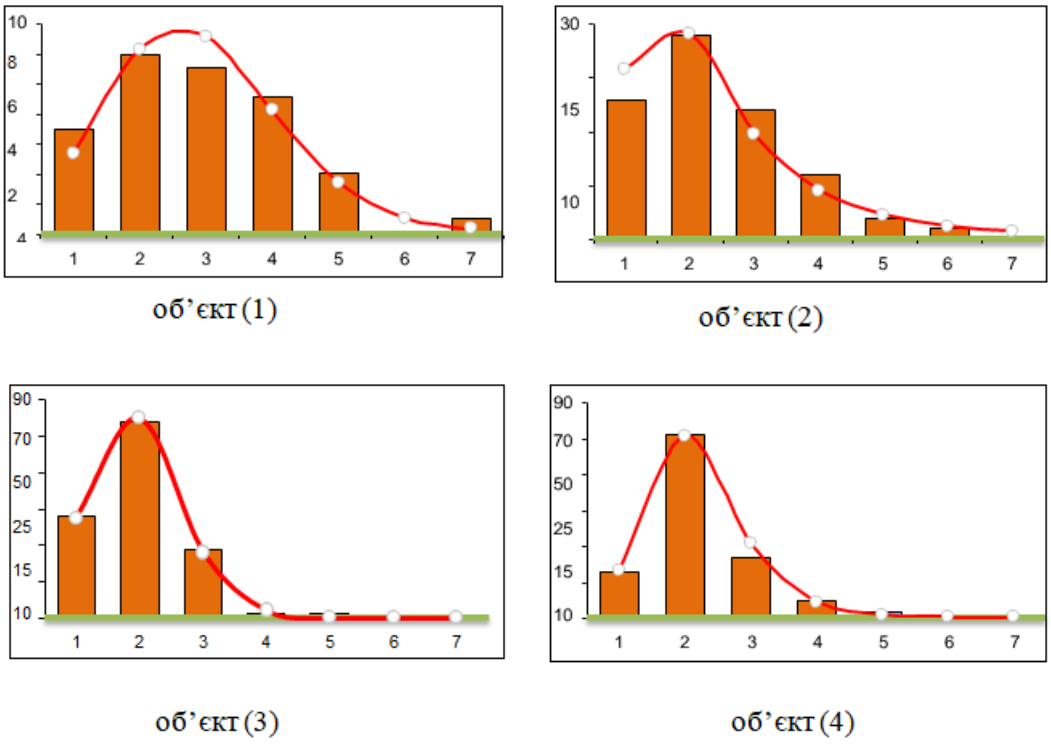

Рисунок 2.6 – Гістограма розподілів для кількості введених команд

При аналізі значень цього параметра необхідно враховувати такі фактори: об’єкт системи, має своє положення у системі. Тому завжди існує значення, ймовірність якого найбільша і складає 0,68...0,89. Дії, пов'язані з роботою об’єкта системи на інших позиціях, можна вважати закономірними.

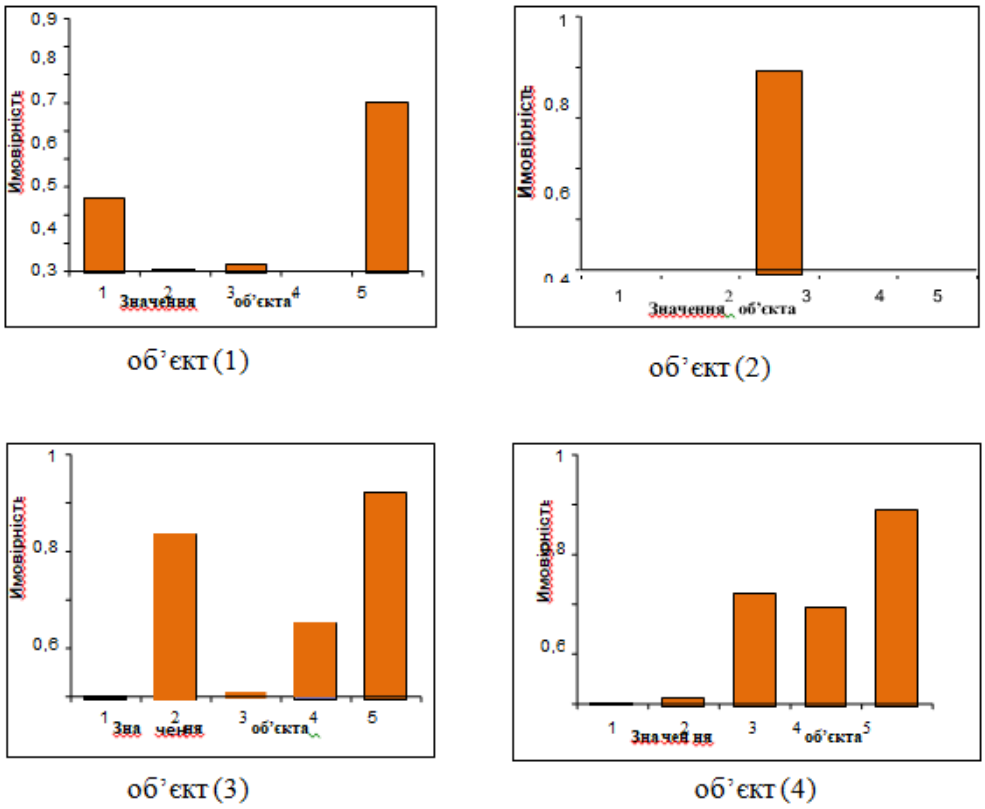

Рисунок 2.7 – Розподіл по різним об’єктам системи

Аналіз значень цього параметра показує, що вони розподілені рівномірно. Значення критерію в середньому рівне 3,46 (при значеннях параметрів $r = 4$, $k = 7$, $s = 3$) забезпечує в середньому 70% відповідність гіпотези реальним даним. Якнайкраще значення критерію в середньому рівне 0,89, для цього параметра було отримано для нормального розподілу. При $r = 4$, $k = 7$, $s = 3$ це значення в середньому забезпечувало 92% відповідність гіпотези реальним даним. На схемі приведено емпіричні та теоретичні розподіли.

2.3.4 Створення генеративної моделі

На основі отриманих залежностей було розроблено програму, яка моделює роботу об’єктів комп'ютерних систем під час сеансу.

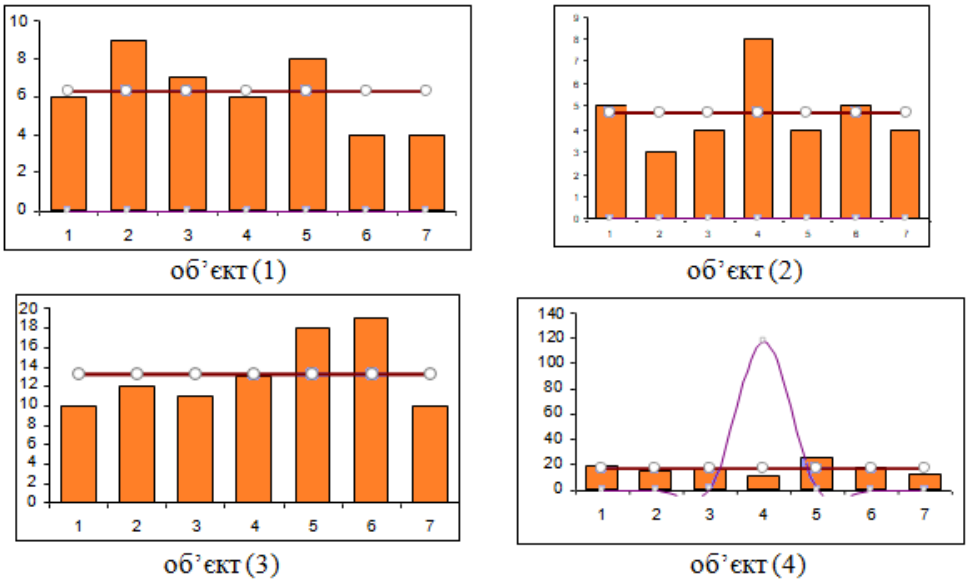

Рисунок 2.8 – Гістрограма розподілу тривалості сеансу

Призначений для об'єктів інтерфейс даної програми дозволяє проводити кількість сеансів, за які необхідно згенерувати дані, загальне число модулів та параметри розподілів складових цієї моделі:

- середнє значення та дисперсію для логарифмічного розподілу (тривалість сеансу);
- ймовірність модуля, за яким об'єкт працює найчастіше (індекс модуля);
- межі рівномірного розподілу (тривалість сеансу);
- середнє значення та дисперсію для нормального розподілу (час початку сеансу).

Розроблений додаток дозволяє генерувати навчальну множину для сеансової моделі об'єкта. Імовірнісні характеристики генерації даних відповідають статистичним показникам, отриманим на представлених даних. Приклад роботи програми наведено нижче.

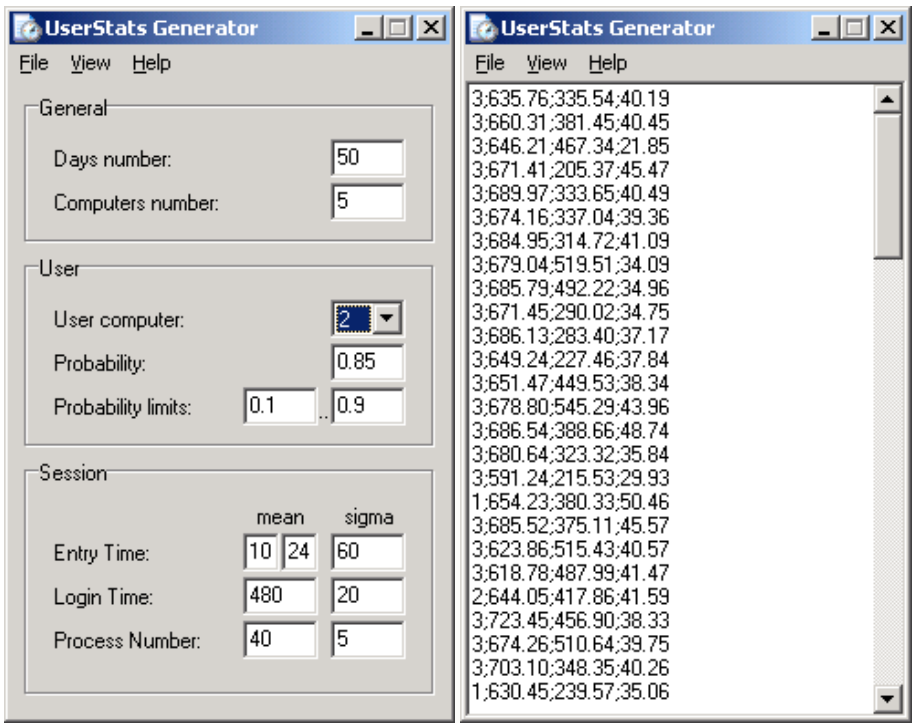

Рисунок 2.9 – Інтерфейс системи генерування даних за сеанс (а) і результати її роботи (б)

Згенеровані за допомогою додатку дані дозволяють оптимізувати структуру нейромережевої моделі і покращити якість функціонування сеансової складової комплексної моделі поведінки.

У даній програмі запропонована комплексна модель поведінки об’єктів систем, яка містить три компоненти: інтерактивний модуль, сеансовий модуль та аналізатор трендів.

Інтерактивна складова базується на прогнозуванні дій об’єктів на основі попередніх дій системи, що дозволяє описати динамічні властивості їх поведінки. Для прогнозування використовується нейромережевий підхід. Це пояснюється тим, що поведінкою об’єктів є нелінійний динамічний процес, а також необхідністю прогнозування в режимі реального часу на основі представлених даних. У основу інтерактивної моделі поведінки об’єктів покладена нейронна мережа прямого розповсюдження. На основі кількості команд, які були правильно спрогнозовані нейронною мережею, робиться висновок про те, чи відповідає поточна поведінка об’єктів раніше

побудованій моделі (тобто чи є поведінка нормальною або дивною).

Слід виділити такі переваги нейромережевих складових комплексної моделі:

- незалежність від кількості об’єктів в системі;
- можливість виявлення прихованих закономірностей в поведінці об’єктів;
- адаптація до зміни поведінки об’єктів.

Запропонований підхід враховує статистичні параметри поведінки об’єктів, які були отримані впродовж сеансу, а саме:

- кількість введених команд;
- результати інтерактивної моделі;
- набір модулів;
- тривалість сеансу;
- час початку сеансу.

Цей набір даних використовується в якості вхідних даних для виявлення нормальної або нестандартної роботи об’єктів за сеанс в цілому. Як і у випадку інтерактивної моделі, для реалізації моделі використовується нейронна мережа прямого розповсюдження. В даному випадку нейронна мережа працює як класифікатор.

Оскільки об’єктам систем властиво змінювати свою поведінку з часом, для адаптації до цього нейромережевих складових важливо забезпечити модуль, який виявлятиме ці зміни.

У атестаційній роботі запропоновано використовувати модуль аналізу трендів, який дозволяє виявляти «тренди» поведінки і надає додаткове підтвердження нестандатрної поведінки об’єктів. Його функціонування засноване на використанні інформації про команди, що вводяться за сеанс і їх порівняння з даними попередніх сеансів. Таким чином, використання модуля аналізу трендів дозволяє відрізнити природну зміну поведінки об’єкта від дивної, а також забезпечити адаптацію нейромережевих складових до можливих змін.

Також розроблена генеративна модель поведінки об’єкта, що забезпечує репрезентативні набори даних для верифікації і ідентифікації комплексної моделі. Для побудови даної моделі виконувалося статистичне моделювання параметрів сеансової складової для кожного об’єкта, насамперед: кількості команд, що вводяться, за сеанс, індекс модуля, тривалість та час початку сеансу. Емпіричні розподіли апроксимовано теоретично. В якості критерію згоди використовувався критерій χ^2.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

При реалізації комплексної нейромережевої моделі об’єктів систем було враховано такі чинники: як правило, комп'ютерна система є розподіленою та містить велику кількість об’єктів, для кожного з яких будується своя модель. При цьому з метою зменшення навантаження на КС модель поведінки доцільно реалізовувати як автономний модуль, що має також можливість змінювати положення в системі, оскільки об’єкт може працювати з різними модулями. Всі ці чинники визначають ті властивості, якими повинна володіти система моніторингу діяльності об’єктів, що розробляється. До них відносяться розподіленість, масштабованість, можливість роботи з різними ОС та форматами даних, автономність і мобільність компонентів (модулів). Цим вимогам краще всього задовольняє агентну структуру реалізації розподілених систем.

3.1 Агентна технологія

У даному розділі агентний підхід використовується для реалізації системи моніторингу діяльності на основі комплексної моделі, описаної раніше. Застосування агентної технології для реалізації даної системи обумовлене наступним: необхідність створення великої кількості моделей (для кожного об’єкта потрібно створити дві відповідні нейромережеві моделі), необхідність мобільності (агент переміщається на клієнтський модуль), взаємодія з БД для навчання [19].

У атестаційній роботі в якості базового визначення агента вибрано наступне твердження: агент – це масив, який може приймати інформацію із зовнішнього середовища і реагувати на зовнішні впливи. Введемо формальний опис агента. У загальному випадку агент визначається набором (множиною):

<S, Prog, Eff, Arch, P, A, G, E>,

- де S (sensors) –множина входів для сприймання інформації;
- Prog (program) – функція, що визначає залежність реакції агента від вхідних дій;
- Eff (effectors) – множина виходів впливу на зовнішнє середовище;
- Arch (architecture) – фізична оболонка для об'єднання базових елементів;
- P (percepts) – інформація отримана агентом;
- A (actions) – реакція агента;
- G (goal) – ціль агента;
- E (environment) – зовнішнє середовище функціонування агента.

Набір S, Prog, Eff, Arch визначає базову конструкцію агента, інакше каркас, а масив P, A, G, E його зміст [19]. Більшість систем виділяють такі властивості, якими повинні володіти агенти: автономність, реактивність, активність, здатність взаємодіяти з іншими агентами чи людьми за допомогою мови спілкування агентів (ACL Agent Communication Language), мобільність (здатність переміщення), адаптивність (здатність до навчання).

У роботі для реалізації системи моніторингу використано програмні агенти (software agents). Їх відмінність полягає в тому, що вони є комп'ютерними програмами і функціонують в комп'ютерних системах. Для програмного агента: *E=комп'ютерна система*, *Arch=програма* (код), а *S* і *Eff* – деякі функції, які в загальному випадку можуть також бути додаткаии, за допомогою яких відбувається обмін інформацією із зовнішнім середовищем.

3.2 Система моніторингу на основі нейромережевої моделі

Оскільки комплексна модель об'єкта враховує динамічні та статистичні властивості поведінки об'єктів в даній системі моніторинг здійснюється як в режимі реального часу (on-line), так і в автономному режимі (off-line).

Метою моніторингу в режимі реального часу є виявлення дефективної діяльності під час роботи об’єкта на основі інтерактивної моделі. Автономний моніторинг призначений для виявлення нехарактерної діяльності об’єкта за сеанс в цілому і виявлення можливих трендів в його роботі. Для цього використовуються сеансова модель і модуль аналізу трендів.

3.2.1 Архітектура системи та базові принципи функціонування

Агент, що інкапсулює інтерактивну модель об’єкта, функціонує під час роботи об’єкта в комп'ютерній системі в режимі реального часу і дозволяє виявляти аномалії (відхилення від нормальної поведінки в його діяльності.

Система моніторингу діяльності об’єктів комп'ютерних систем містить базові компоненти:

- агент, що інкапсулюється, включає в себе та приховує інтерактивну модель об’єкта, також працює в режимі реального часу (On-line User Agent);
- агент, що інкапсулює сеансову модель об’єкта і модуль аналізу трендів працює в автономному режимі (Off-line User Agent);
- агент-контролер (Controller Agent) це наглядач роботи інших агентів всистемі;
- БД (Database), яка містить дані та параметри існуючих моделей.

Розглянемо кожен з компонентів запропонованої системи моніторингу детальніше [9].

У виразі цей тип агента характеризується такими параметрами (з урахуванням того, що в цьому випадку параметри *E*, *Arch*, *S* і *Eff* для програмного агента фіксовані):

- G – виявлення аномальної діяльності об’єкта під час його роботи;
- P та A – послідовність команд, які введені об’єктом;
- Prog: P–>A – збір даних про діяльність об’єктів, прогнозування команд з використанням нейронної мережі та обчислення значення (*i).*

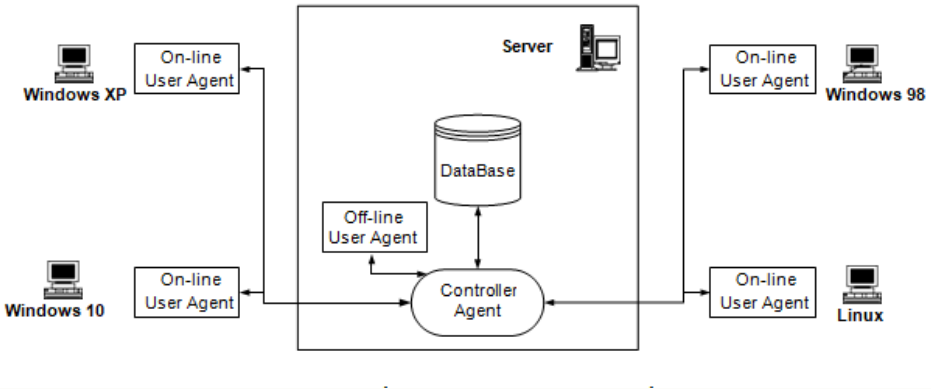

Рисунок 3.1 – Архітектура системи моніторингу

В процесі роботи об’єкта цей агент прогнозує дії, які згодом порівнюються з базовими. Якщо відносна кількість правильно прогнозованих команд впродовж сеансу більша заданого порогу, вважається, що поведінка об’єкта стандартна, в іншому випадку ненормальна. Ще одна функція цього агента полягає в зборі інформації про поведінку об’єкта та її збереження в базі даних. При цьому даний тип агента повинен бути розроблений для різних операційних систем, які використовуються в комп'ютерній системі, наприклад: Win98, Linux, FREEBSD, Win2000/XP [20].

Агент, що інкапсулює сеансові моделі об’єктів, функціонує після завершення сеансу роботи об’єктів та записує наступні значення параметрів набору, який визначається заданими значеннями:

- G – виявлення нехарактерної діяльності об’єктів за сеанс в цілому і виявлення можливих трендів в їх роботі;
- P – статистичні дані про діяльність об’єктів;
- A – ймовірність характерної діяльності об’єктів;
- Prog: P–>A – обчислення ймовірності нетипової діяльності.

Агент-контролер – агент, який відповідає за функціонування системи в цілому і управляє роботою інших агентів.

У позначеннях цей тип агента має такий опис:

- G – управління системою моніторингу;
- P – дані про елементи системи;
- A – організація взаємодії між елементами системи;
- Prog: P–>A – створення різних типів агентів, отримання інформації з БД та перенавчання нейронних мереж, адаптація моделі.

На основі зібраної інформації агент, що інкапсулює сеансову модель об’єктів, уточнює, на скільки діяльність об’єктів була нетиповою, та що саме визначається значенням проміжку [0,1], що вказує на ймовірність стандартної поведінки.

Такий агент отримує інтегральні дані про поведінку всіх елементів системи. За його підтримки виникає взаємодія в рамках системи та контролюється її функціонування, а також відбувається зв’язок між агентами. Агент звертається до БД з метою отримання та передачі інформації. Під його контролем відбувається навчання різних типів агентів. Він також поповнює базу даних новими даними про об’єкти системи. У БД зберігається інформації, яка необхідна для функціонування системи.

Коли об’єкт (Object) входить в систему, а саме починає роботу за модулем Host, агент-контролер (Controller Agent) створює і ініціалізовує відповідний агент, який інкапсулює інтерактивну модель користувача (On-line Object Agent). Отримані параметри моделі об’єкта з бази даних (DB) (параметри нейронної мережі конкретного об’єкта, агент переміщається на сторону об’єкта та починає роботу, виконуючи свої функції по прогнозуванню команд та збір інформації з аудит-файлу (Logger). Якщо виявлена нетипова або нехарактерна робота об’єкта, відповідне повідомлення посилається агентові-контролеру. Після завершення сеансу об’єкта такий агент буде знищено. Блок-схема послідовностей для агента даного типу приведена на рисунку.

Після закінчення операційного циклу, а саме при мінімальному навантаженні на систему, агент-контролер створює та ініціалізує відповідний

агент, що інкапсулює сеансову модель об’єкта (Off-line Object Agent). На основі статистичних даних, отриманих під час сеансу об’єкта, сеансової моделі та модуля аналізу трендів цей агент визначає характер активності об’єкта (нормальна або нетипова) та визначає можливі тренди в його роботі.

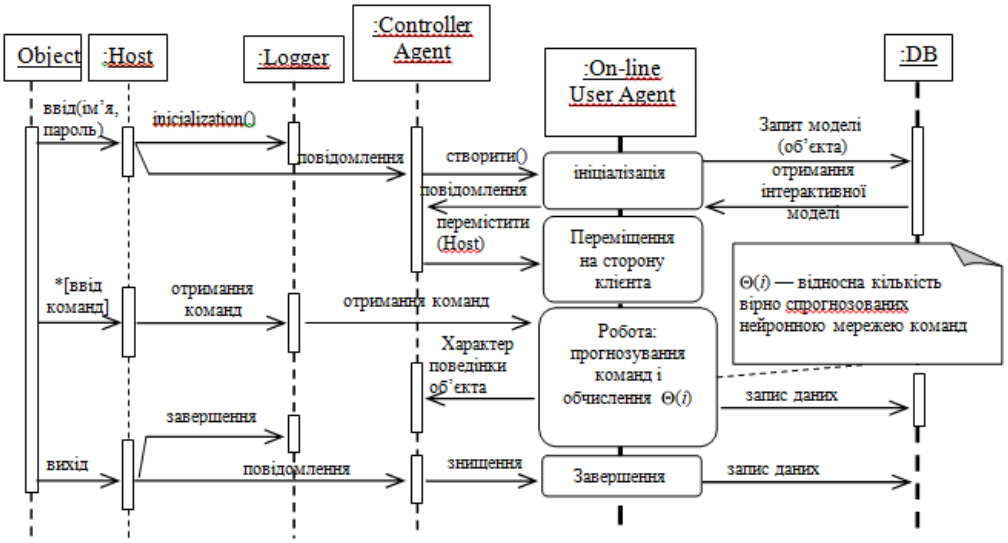

Рисунок 3.2 – UML–діаграма послідовностей роботи агента

У разі нетипової активності відповідна інформація посилається агенту-контролеру. Діаграма послідовностей наведена нижче.

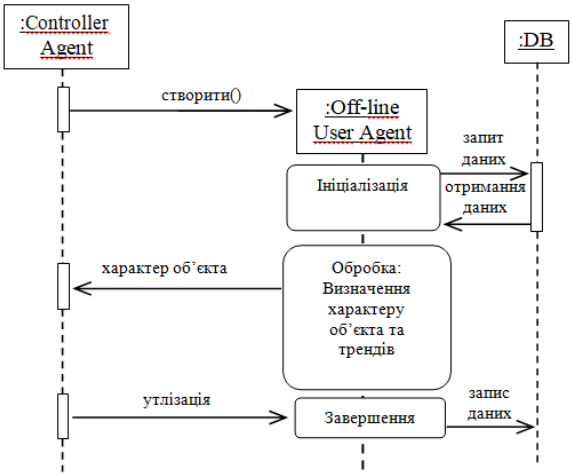

Рисунок 3.3 – UML–діаграма послідовностей роботи агента

3.2.2 Діаграма класів нейромережі

Класи для реалізації нейромережевих моделей згруповані в пакет neural.

Клас ObjectAgent реалізує шаблон інтелектуального агента, що інкапсулює нейромережеві моделі. Клас Layer пакету neural реалізує шар нейронної мережі [19]. Атрибути LSize, Weights і Outputs даного класу визначають відповідно кількість нейронів в шарі, значення вагових коефіцієнтів та вихід нейронів шару. Метод dTrFunction() реалізує активаційну функцію нейронів, а метод Run() призначений для обчислення виходу нейронів.

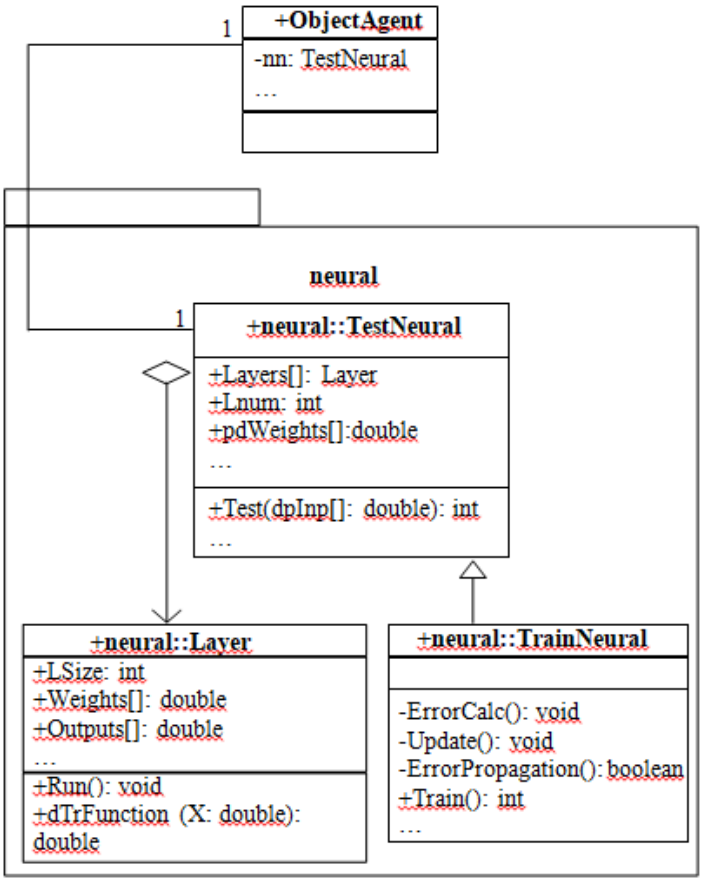

Рисунок 3.4 – UML–діаграма (реалізація) класів нейромережевої моделі

У класі TrainNeural реалізовані методи, спрямовані на самонавчання нейронної мережі. Метод ErrorCalc() використовується для обчислення частоти помилок, а метод Update() для зміни вагових коефіцієнтів. Метод ErrorPropagation() реалізує алгоритм зворотного розповсюдження помилки. Метод Train() є основним, за допомогою якого здійснюється ітераційна зміна вагових коефіцієнтів методом градієнтного спуску.

Клас TestNeural реалізує режим тестування нейронної мережі. Даний клас використовує масив об'єктів типу Layer для реалізації нейронної мережі. Атрибути Lnum і pdWeights задають кількість шарів в нейронній мережі і матрицю вагових коефіцієнтів. Метод Test() обчислює вихід нейронної мережі при подачі вхідного вектора.

Програмний код реалізації класів пакету neural наведено в додатку А.

3.3 Реалізація модуля аналізу

3.3.1 Інструменти реалізації модулів

Система нейромережевого аналізу активності об’єктів системи була реалізована на основі мобільних агентів, які здатні змінювати своє положення в мережі. В якості технології реалізації та середовища програмування мобільних агентів були використані Java та Aglets Software Development Kit (ASDK).

ASDK є програмним забезпеченням, яке випущено компанією IBM та призначене для створення мобільних агентів, які в реалізації ASDK називаються аглетами [19]. Версія ПЗ 2.0.2 має просту та прозору структуру, зручний графічний інтерфейс користувача, який надається сервером аглетів Tahiti та має корисну документацію. Варто виділити такі властивості ASDK:

- мобільність агентів, яка заснована на серіалізації та переміщенні двійкового коду Java з одного місця в інше;
- використання розробленого стандарту MASIF (Mobile Agent System

Interoperability Facility), направленого на забезпечення взаємодії між різними агентами;

- використання на прикладному рівні протоколу ATP (Aglets Transfer Protocol) з підтримкою HTTP-тунелювання (HTTP Tunneling) для переміщення агентів;
- використання базової системи безпеки Java (JDK keytool).

Java як мова програмування агентів представляє великий набір інструментів, що дозволяє спростити розробку багато агентних систем. Слід зазначити такі властивості Java, як платформна незалежність, багато поточне програмування, безпечне виконання кодів, динамічне завантаження класів, серіалізація об'єктів, а саме уявлення у вигляді бінарної послідовності. Також, слід відзначити такі недоліки даної технології, що пов'язані з програмуванням мобільних агентів на Java: відсутність захисту посилань на об'єкти, погана підтримка контролю ресурсів та відсутність підтримки збереження і відновлення стану виконання.

Для більшого розуміння було розглянуто принципи створення аглетів з використанням ASDK.

3.3.2 Реалізація аглетів у моделі

Базова функціональність мобільних агентів визначається програмним інтерфейсом Aglet API (Application Programming Interface). Нижче показані основні інтерфейси та класи Aglet API і взаємозв'язки між ними.

У загальному випадку аглети – це об'єкти Java, які можуть змінювати положення між різними модулями мережі. Під переміщенням аглета мається на увазі передача програмного коду із збереженням стану агента, значень атрибутів. Отже, якщо агент виконує деякі дії з одним модулем, то у будь-який момент він може припинити процес та перемістится на інший і продовжити обробку.

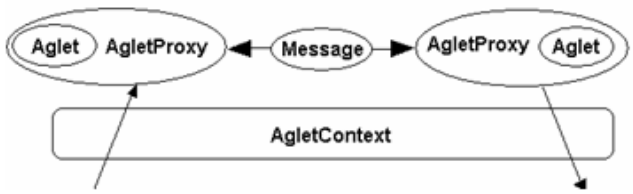

Рисунок 3.5 – Базові інтерфейси та класи Aglet API

Далі опишемо основні призначення інтерфейсів та класів та інтерфейсу Aglet API.

Клас com.ibm.aglet.Aglet є базовим та визначає основні атрибути та методи, які повинен містити мобільний агент. При цьому будь-який створюваний аглет повинен наслідувати цей абстрактний клас.

Інтерфейс com.ibm.aglet.AgletContext призначений для створення середи (runtime environment), в рамках якої будуть існувати агенти.

Інтерфейс com.ibm.aglet.AgletProxy призначений для створення об'єкту-посередника самому аглету. Тобто, будь-який запит, адресований агенту для виконання будь-яких дій, здійснюватиметься через посередника, який базується на установці диспетчера безпеки (SecurityManager) та визначає, які методи аглета можна викликати.

Клас com.ibm.aglet.Message призначений для створення об'єктів, через які здійснюється спілкування між агентами.

Раніше було зауважено, що будь-який агент, що створюється об’єктом, повинен успадковувати клас com.ibm.aglet.Aglet або його реалізацій. Нижче наведено методи такого класу, які мають бути перевизначені:

- void run() – метод викликається кожен раз при створенні чи переміщенні, активації або клонуванні, створення ідентичної копії аглета;
- void onCreation(Object init) – метод викликається один раз при створенні нового аглета;
- void onDisposing() – метод викликається при знищенні аглета.

Нижче наведено фрагмент коду для створення нового аглета:

```
import com.ibm.aglet.*; // Підключення пакету com.ibm.aglet

public class ObjectAglet extends Aglet { public void onCreation(Object
init){ System.out.println("Aglet is created!");


public void run() { System.out.println("Hello!");


public void onDisposing() { System.out.println("bye!");
}
}
```

Приклад 3.1 – Функція Void onCreation

Щоб забезпечити мобільність аглетів, а саме можливість переміщення на віддалений модуль, в ASDK використовується модель подій (event model). Так, при відправці або прибутті агента на інший модуль, генерується відповідна подія, що пов'язана з одним з методів інтерфейсу com.ibm.aglet.event.MobilityListener, адже він відповідає за обробку подій, пов'язаних з мобільністю аглетів [19]. Створивши клас, що реалізовує цей інтерфейс, об’єкти можуть перевизначити потрібні методи. Наведемо приклад:

```
import com.ibm.aglet.Aglet;
import             com.ibm.aglet.event.MobilityEvent;             import
com.ibm.aglet.event.MobilityListener;
class   ObjectListener   implements   MobilityListener{   //   Реалізація
інтерфейсу MobilityListener
public void onDispatching(MobilityEvent event){
// Цей метод викликається перед переміщенням аглета
}
public void onArrival(MobilityEvent event){
// Цей метод викликається після прибуття на новий хост
}
public void onReverting(MobilityEvent event){
}}
public class ObjectAgletextends Aglet { public void onCreation(Object
init){
MobilityListener   listener   =   new   ObjectListener();   //   Створення
екземпляра для користувача класу ObjectListener
addMobilityListener(listener); // Зв'язує обробник подій з listener}}
```

Приклад 3.2 – Функція event.MobilityListener

3.3.3 Програмна реалізація та результат тестування

Запропонована система обслуговування була реалізована на основі клієнт – серверної архітектури. Серверна частина є спеціальним додатком (серверною платформою), до функцій якого відносяться створення та розміщення агентів системи, які використовуються тут, перерозподіл запитів та взаємодія з БД, а клієнтська частина призначена для забезпечення роботи агентів на призначеному модулі дослідження. Її функції полягають в розміщенні агентів на стороні об’єктів та ведення аудиту про їх діяльність. Також реалізовано графічний інтерфейс наведений нижче, в якому відображається інформація про позначення об’єкта та задіяну ОС, про параметри клієнтської платформи, про стан агента і дані про роботу агента.

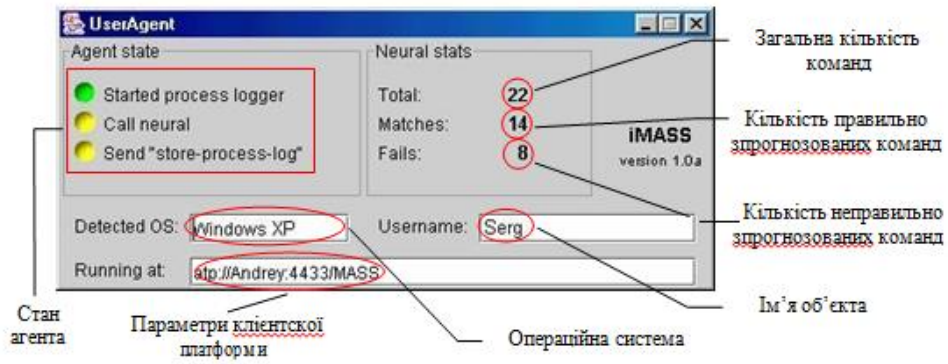

Рисунок 3.6 – Графічний інтерфейс користувача

Взагалі, задача оцінки ефективності роботи багатоагентної системи моніторингу діяльності об’єктів систем є достатньо складною задачею та вимагає ретельного дослідження. У даній роботі застосовано наступний простий підхід. Відомо, що близько 70-85% атак ініціюється користувачами всередині комп'ютерної системи. Пропонована багатоагентна система моніторингу дозволяє в середньому в 77% випадків виявляти нетипову поведінку об’єктів. Визначено, що 15-26% типів атак успішно виявляються

іншими засобами. Тоді ймовірність виявлення атаки складає приблизно 0,7. Якщо вважати, що існуючі методи та підходи виконання безпеки (такі як системи виявлення та запобігання вторгненням, антивірусні програми, різноманітні засоби адміністратора) дозволяють виявляти та запобігти в середньому 43% атакам, то використання багатоагентної системи аналізу поведінки об’єктів в мережах дозволяє збільшити її захищеність від внутрішніх загроз в середньому на 37% [2].

У даному розділі розглянуті питання, пов'язані з програмною реалізацією комплексної моделі об’єктів систем. Для реалізації системи був використаний агентний підхід. Це дало можливість використовувати запропоновану систему дослідження в гетерогенному середовищі (з різними операційними системами та форматами даних), а також забезпечити автономність та мобільність компонентів (можливість переміщення моделі на сторону об’єкта). Використання агентної технології дозволило зробити систему розподіленою та масштабованою. Так, при додаванні в систему нового об’єкта необхідно тільки внести до БД відповідну інформацію про модель об’єкта, а все інше система налаштує самостійно.

Проте відзначимо, що статистика ефективності засобів виявлення вторгнень та нетипової поведінки відсутня, тому оцінки є приблизними, а саме питання оцінки ефективності засобів інформаційної безпеки вимагає окремого опрацьовування.

В якості технології реалізації та середовища програмування мобільних агентів були вибрані, відповідно, Java і Aglets Software Development Kit (ASDK). Реалізована нейромережева система дослідження підтвердила дієздатність та ефективність запропонованої моделі об’єкта.

4 ІНСТРУКЦІЯ КОРИСТУВАЧА

Головним призначенням розробленої системи обслуговування є створення та розміщення агентів системи, напрямлені на перерозподіл запитів та взаємодія з БД, а клієнтська частина призначена для забезпечення роботи агентів на призначеному модулі дослідження. На меті стоїть визначення ефективних механізмів виявлення нетипових запитів у роботі користувача компьютерної системи та забезпечення інформаційної безпеки.

Перед початком використання додатку, зазначимо, що система реалізована на клієнт – серверній архітектурі, насамперед серверна частина є додатком, що потребує завантаження додаткового ПО. Також нема потреб до конфігурації комп'ютера користувача, що значно полегшує інсталяцію додатка.

Для роботи з системою користувач повинен мати прості навички з використання ОС Windows та елементарні знання про роботу клієнт-серверних застосувань.

Підготовка системи до роботи є одним з важливих етапів, тому що спочатку треба завантажити необхідне устаткування для системи. Для роботи web - сервера необхідно завантажити систему сервер MySQL з офіціального сайту (https://www.mysql.com/downloads/). Треба уважно вибрати потрібну версію дистрибутиву для вашої ОС. Обробка та обмін даними буде відбуватися через сервер-модуль [15].

Після скачування запустіть устаткування та увімкніть сервер Mysql для роботи та згорніть застосування на робочому столі.

Далі необхідно запустити додаток Aglets у якому буде відбуватися робота системи. У рядок URL необхідно вписати локальну адресу вашого хосту і шлях до виконуючих файлів додатка, наприклад: /localhost/oleg…./.

При виконанні всіх вищезазначених моментів, система буде повністю готова до роботи.

Інтерфейс програми простий, зрозумілий, немає багаторівневої системи вкладень меню тощо.

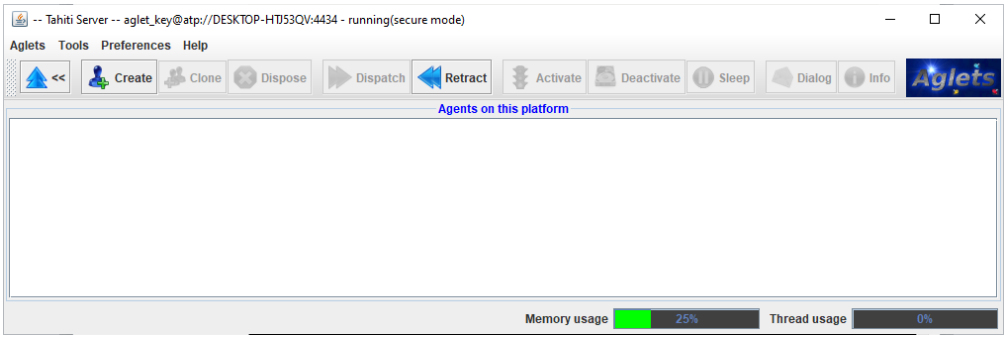

Рисунок 4.1 – Вікно програми для створення агентів

Перед початком роботи з системою є доречним прочитати вкладку «Help» з докладною інформацією про користування даним застосуваннях. Інформаційна сторінка має такий вигляд (рисунок 4.2).

Рисунок 4.2 – Інформаційна сторінка

Інформаційне повідомлення містить в собі деякі дані про створене застосування та правила користування ним у домашніх умовах: «SourceForge

uses markdown syntax everywhere to allow you to create rich text markup, and extends markdown in several ways to allow for quick linking to other artifacts in your project. Markdown was created to be easy to read, easy to write, and still readable in plain text format….». Далі натиснувши кнопку «Create» відкривається наступне вікно для роботи. Тут можна побачити такі функції, як «Add to the list» та «Remove from the list» і звісно рядок для запису URL та Code Base.

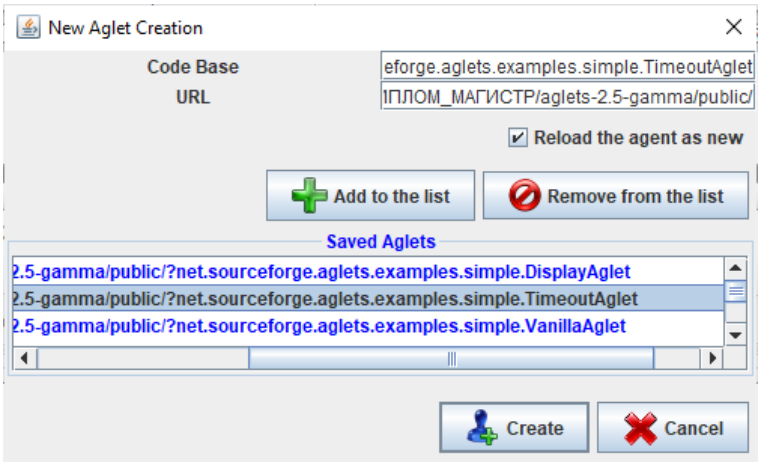

Рисунок 4.3 – Вікно для додавання агенту

Система пропонує нам додати URL-адресу. Для цього просто беремо нашу URL - адресу та вставляємо у форму. При запуску програми також працює вікно для виклику системних команд, завдання параметрів мережі, створення аглетів, тощо.

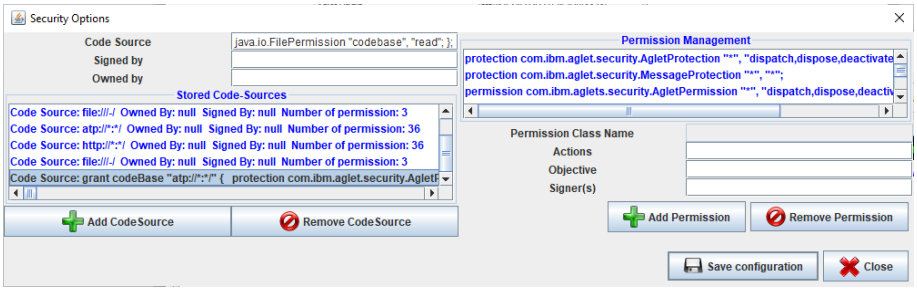

Рисунок 4.4 – Вікно налаштування агенту

Далі треба натиснути кнопку «Add Code Source», а за нею кнопку «Save configuration». Звісно, якщо Ви хочете виконати перевірку лише однієї адреси, тоді дійте саме так, але якщо ви збираєтесь виконати перевірку декількох адрес, тоді по черзі додайте їх у форму за наступним алгоритмом:

- а) додайте першу адресу, натисніть «Add Code Source»;
- б) додайте другу адресу і також натисніть «Add Code Source»;
- в) повторіть це стільки разів, скільки адрес Вам необхідно сканувати (рисунок 4.4).

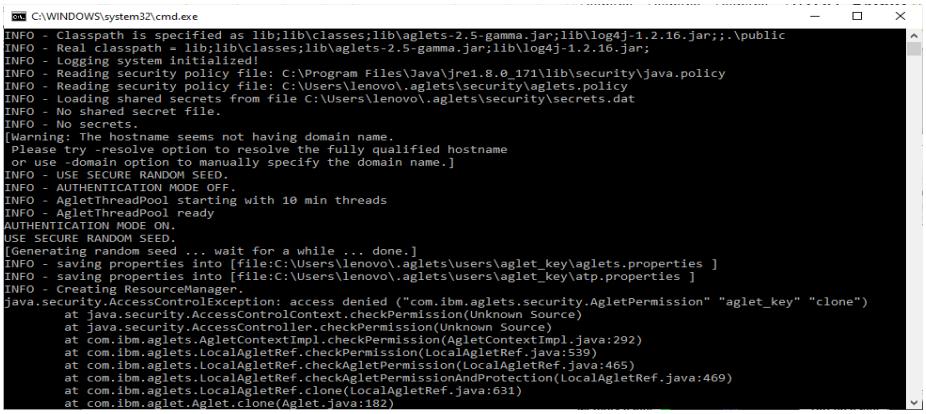

Рисунок 4.5 – Командне вікно Aglets

Коли всі налаштування виконано, можна розпочати процес модуляції. Для цього натискаємо кнопку «Network Preferences» та деякий час очікуємо відповідь від сервера. Як правило, очікування займає не більше 2-3 хвилин і тільки при великій кількості запитів. Як тільки обробка буде закінчена, нижче випаде вікно з інформацією про виконання пошуку нетипових запитів у діях об’єктів. Будуть наведені такі дані, як: кількість створених запитів, кількість спрогнозованих команд, кількість невірно спрогнозованих команд, статус обробки запитів у вигляді індикатора та параметри клієнтської платформи.

Нижче наведено приклад виведеної інформації щодо обробки запиту (рисунок 4.6).

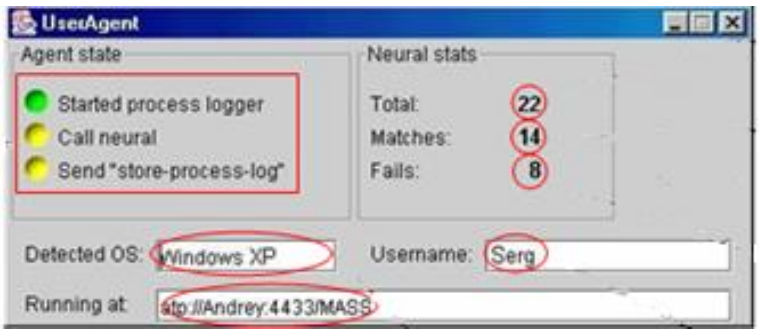

Рисунок 4.6 – Результат модуляції

На цьому робота з системою закінчена. Достатньо надалі закрити додаток та вимкнути БД.

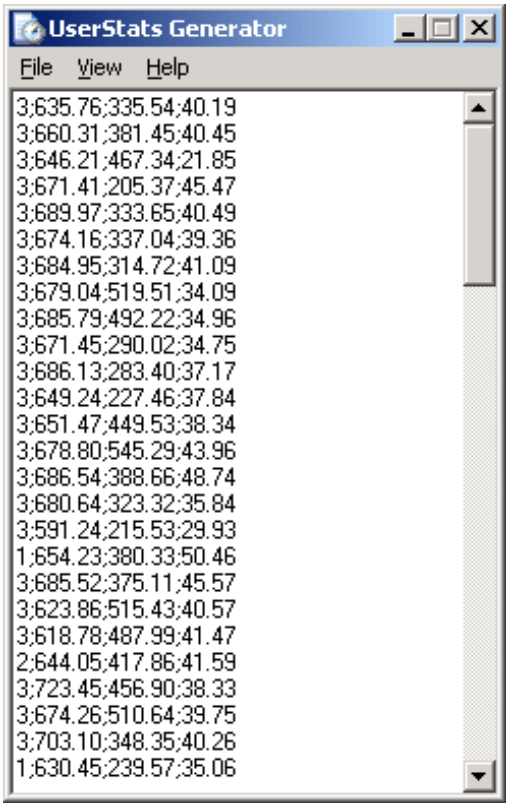

Рисунок 4.7 – Результат статичних показників

Також при роботі з системою трапляються аварійні ситуації, але це нормально при напруженій роботі додатка. Нижче приведено клонування агенту.

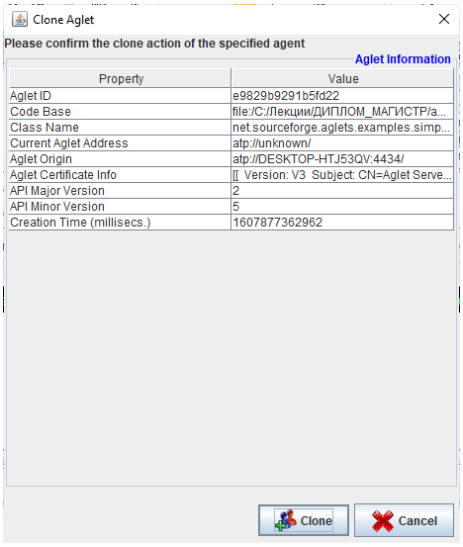

Рисунок 4.8 – Клонування агента

Насамперед для налаштування мережі використовуються параметри мережі. Налаштовуємо запит HTTP - тунелювання та Proxy port. Також треба додати «Create a new shared secret» та натиснути «Apply», щоб додати модуль агента.

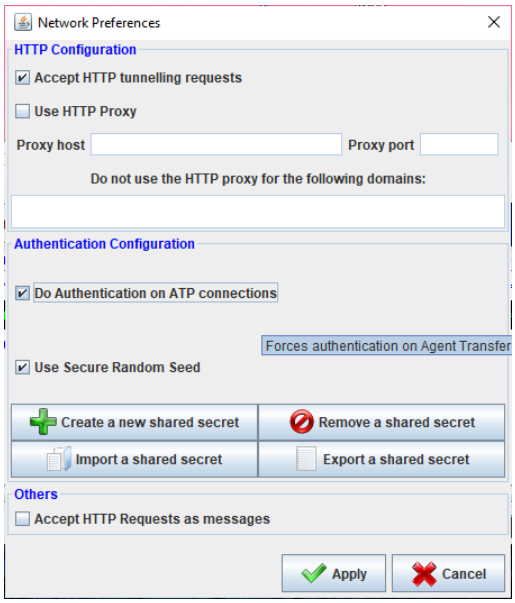

Рисунок 4.9 – Налаштування мережевих параметрів

ВИСНОВКИ

Дана атестаційна робота була присвячена розробці моделей аналізу поведінки об’єктів систем з метою їх застосування в системах дослідження та виявлення нестандартної діяльності. У роботі запропонована комплексна модель поведінки об’єктів, побудована генеративна модель та реалізована комплексна модель у вигляді багатоагентної системи виявлення нетипових дій в поведінці об’єктів систем.

Проведено аналіз існуючих методів побудови моделей поведінки та виявлено не ефективність існуючих підходів. Так, в більшості робіт враховують тільки динамічні властивості поведінки об’єктів, або тільки статистичну інформацію. При цьому ні як не враховують можливі тренди зміни поведінки об’єктів. Також розроблена комплексна нейромережева модель поведінки у системах, що включає три компоненти: інтерактивний блок (враховує динаміку поведінки), сеансів блок (враховує статистичні властивості) та модуль аналізу трендів (виявлення можливих зміни в поведінці).

На відміну від існуючих підходів, запропонована модель дозволяє забезпечити комплексний підхід до аналізу поведінки у режимі реального часу та у відкладеному режимі після закінчення сеансу. Розроблена генеративна модель поведінки забезпечує репрезентативні набори даних для ідентифікації та верифікації комплексної моделі. Для побудови виконувалося статистичне моделювання даних. В якості критерію відповідності гіпотези реальним даним використовувався критерій χ^2. На основі отриманих розподілів згенерований необхідний набір даних, що дало можливість оптимізувати архітектуру нейронної мережі для сеансової моделі та покращило її функціонування.

Розроблена об'єктно-орієнтована модель представлення нейромережевих моделей об’єкта в рамках агентної парадигми.

Запропоновані принципи функціонування та взаємодії агентів, що реалізують нейромережеві моделі поведінки об’єктів. Агентна реалізація дозволила використовувати розроблену систему в гетерогенному середовищі, забезпечити автономність та мобільність компонентів, дозволила зробити систему розподіленою та легко масштабованою. Використання багатоагентної системи аналізу поведінки об’єктів в мережах дозволяє збільшити її захищеність в середньому на 37%.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Основы теории искусственных нейронных сетей О.Г. Руденко, Е.В. Бодянский - Харьков: Телетех, 2002

2. Ивахненко А.Г. Самообучающиеся системы распознавания и автоматического управления. – К.: Техника, 1969. – 245 с.

3. Растригин Л.А., Пономарев Ю.П. Экстраполяционные методы проектирования и управления. – М: Машиностроение, 1986. – 120 с.

4. Горелик А.Л., Скрипкин В.А. Методы распознавания: Учеб. пособие. – М.: Высш. шк., 1984. – 208 с.

5. Stolfo S.J., Fan D.W., Lee W., Prodromidis A.L. Credit card fraud detection using meta-learning: Issues and initial results // In Proc. Of AAAI-97 Workshop on AI Approaches to Fraud Detection & Risk Management, AAAI Press. – 1997. – P. 83– 90.

6. Принципы организации систем нечеткого регулирования на однородных локально-параллельных алгоритмах О.Ф. Михаль, О.Г. Руденко - Управляющие системы и машины, 2001

7. Russel, S., Norvig, P. Artificial Intelligence: A Modern Approach. – Upper Saddle River NJ: Prentice Hall, 1995. – 932 p.

8. Luck M., McBurney P., Preist C. Agent Technology: Enabling Next Generation Computing. – N.Y.: AgentLink, 2003. – 94 p.

9. Haykin S. Neural Networks: a comprehensive foundation. – Upper Saddle River, New Jersey: Prentice Hall, 1999. – 842 p.

10. Cover T.M., Hart P.E. Nearest neighbor pattern classification // IEEE Transactions on Information Theory. – 1967. – vol. IT-13. – P. 21–27.

11. Aglets Software Development Kit. – http://sourceforge.net/projects/aglets/.

12. Oshima M., Karjoth G. Aglets Specification (1.0) – http://www.trl.ibm.com/aglets/spec10.htm.

13. Ночевнов Д.П. Метод адаптивного информационного поиска на основе контекстной модели пользователя // Вісник Черкаського державного технологічного університету. – 2003. – № 3. – С. 17-23.

14. Куссуль Н., Соколов А. Адаптивное обнаружение аномалий в поведении пользователей компьютерных систем с помощью марковских цепей переменного порядка. Часть 1. Адаптивная модель марковских цепей переменного порядка // Проблемы управления и информатики. – 2003. – № 3. – С. 83-93.

15. Куссуль Н.Н., Соколов А.М. Адаптивное обнаружение аномалий в поведении пользователей компьютерных систем с помощью марковских цепей переменного порядка. Часть 2. Методы обнаружения аномалий и результаты экспериментов // Проблемы управления и информатики. – 2003. – № 4. – С. 83– 88.

16. Tauscher L., Greenberg S. How people revisit Web pages: Empirical findings and implication for the development of history systems // International J. of Human Computer Studies. – 1997. – 47(1). – P. 97-138.

17. Debevc M., Meyer B, Svecko R. An adaptive short list for documents on the world wide web // Proc. of the 1997 International Conf. on Intelligent User Interfaces. – Orlando, FL (USA). – 1997. – P. 209-211.

18. Masui T., Nakayama K. Repeat and Predict – Two Keys to Efficient Text Editing // In Proc. of CHI'94. – 1994. – P. 118-123.

19. Гаскаров Д.В., Голинкевич Т.А., Мозгалевский А.В. Прогнозирование технического состояния и надежности радиоэлектронной аппаратуры. – М.: Сов. Радио, 1974. – 224 с.

20. Блинов И.Н., Гаскаров Д.В., Мозгалевский А.В. Автоматический контроль систем управления. – Л.: Энергия, 1968. – 165 с.