https://doi.org/10.31891/2219-9365-2025-81-34
УДК 004.491.42

ЧАЙКОВСЬКИЙ Максим
Хмельницький національний університет
https://orcid.org/0000-0002-9596-6697
e-mail: max.chaikovskyi@gmail.com

АРХІТЕКТУРА ТА ЗАСОБИ МУЛЬТИАГЕНТНОЇ СИСТЕМИ ВИЯВЛЕННЯ ПОЛІМОРФНИХ ВІРУСІВ В КОМП'ЮТЕРНИХ МЕРЕЖАХ

У роботі вперше розроблено архітектуру мультиагентної системи виявлення поліморфних вірусів, яка каскадно делегуючи повноваження, здійснює аналіз середовища, досліджує темпи поширення, виявляє, класифікує поліморфні віруси та використовує різні стратегії прийняття рішення, враховуючи типи загроз. В структуру даної системи входять інтелектуальні агенти (ІА): ІА «Аналіз», ІА «Темп поширення», ІА «Виявлення», ІА «Класифікація», ІА «Прийняття рішення», які виконують встановлені ролі та взаємодіють між собою. ІА «Аналіз» перевіряє систему на наявність підозрілих або шкідливих дій, аналізуючи файлову систему, оперативну пам'ять, мережевий трафік та поведінку процесів; включає збір інформації, що стосується програмного забезпечення, та проведення глибокого аналізу для визначення характеру загрози, її джерела та можливого впливу. ІА «Темп поширення» здійснює моделювання процесу поширення поліморфних вірусів на основі моделі Лотки-Вольтерра (в якій α – ймовірність того, що кількість поліморфних вірусів зростатиме; β – ймовірність того, що поліморфні віруси різних рівнів складності будуть виявлені за допомогою обраних методів, технологій та інструментів; γ – ймовірність того, що деякі з обраних методів, технологій та інструментів не будуть ефективними при виявленні поліморфних вірусів різних рівнів складності в результаті появи нових їх різновидів; δ – ймовірність того, що поліморфні віруси різних рівнів складності вимагатимуть комплексного використання обраних методів, технологій та інструментів, а також новітніх підходів) та визначає, які методи виявлення поліморфних вірусів слід використати. ІА «Виявлення» використовує комплекс методів для виявлення поліморфних вірусів (кількість методів визначена ІА «Темп поширення»): алгоритми пошуку рядка, інтелектуальний аналіз даних, аналіз в пісочниці, машинне навчання, метод розробки структурних функцій, PLN. ІА «Класифікація» здійснює нечітку класифікацію виявлених поліморфних вірусів (ІА «Виявлення») згідно 6 рівнів їх складності за алгоритмом: визначення характеристик виявлених поліморфних вірусів та формування дерева логічного висновку; опис лінгвістичних змінних; визначення функцій належності лінгвістичних термів; формування бази знань системи нечіткого висновку; визначення ймовірності належності досліджуваного файлу до поліморфних вірусів різних рівнів складності; нечітка класифікація поліморфних вірусів. ІА «Прийняття рішення» використовує різні стратегії прийняття рішення (оповіщення, блокування, карантин, видалення), враховуючи типи загроз.

Ключові слова: поліморфний вірус, інтелектуальний агент, мультиагентна система, модель Лотки-Вольтерра, система нечіткого логічного висновку, методи виявлення поліморфних вірусів.

CHAIKOVSKYI Maksym
Khmelnytskyi National University

ARCHITECTURE AND MEANS OF A MULTIAGENT SYSTEM FOR DETECTION OF POLYMORPHIC VIRUSES IN COMPUTER NETWORKS

The work first developed the architecture of a multi-agent system for detecting polymorphic viruses, which, by cascading delegation of authority, analyzes the environment, investigates the rate of spread, detects, classifies polymorphic viruses, and uses various decision-making strategies, taking into account the types of threats. The structure of this system includes intelligent agents (IA): AI «Analysis», AI «Spread Rate», AI «Detection», AI «Classification», AI «Decision-Making», which perform established roles and interact with each other. AI «Analysis» checks the system for suspicious or malicious actions by analyzing the file system, RAM, network traffic, and process behavior; includes collecting information related to software and conducting in-depth analysis to determine the nature of the threat, its source, and possible impact. The «Spread Rate» AI models the process of spreading polymorphic viruses based on the Lotka-Volterra model (in which α is the probability that the number of polymorphic viruses will increase; β is the probability that polymorphic viruses of different levels of complexity will be detected using the selected methods, technologies and tools; γ is the probability that some of the selected methods, technologies and tools will not be effective in detecting polymorphic viruses of different levels of complexity as a result of the emergence of their new varieties; δ is the probability that polymorphic viruses of different levels of complexity will require the complex use of the selected methods, technologies and tools, as well as the latest approaches) and determines which methods for detecting polymorphic viruses should be used. The «Detection» AI uses a set of methods for detecting polymorphic viruses (the number of methods is determined by the «Spread Rate» AI): string search algorithms, data mining, sandbox analysis, machine learning, structural function development method, PLN. The «Classification» AI performs a fuzzy classification of detected polymorphic viruses («Detection» AI) according to 6 levels of their complexity according to the algorithm: determining the characteristics of detected polymorphic viruses and forming a logical inference tree; describing linguistic variables; determining the membership functions of linguistic terms; forming a knowledge base of the fuzzy inference system; determining the probability of the studied file belonging to polymorphic viruses of different levels of complexity; fuzzy classification of polymorphic viruses. The «Decision-Making» AI uses different decision-making strategies (alert, block, quarantine, delete), taking into account the types of threats.

Keywords: polymorphic virus, intelligent agent, multi-agent system, Lotka-Volterra model, fuzzy logic inference system, methods for detecting polymorphic viruses.

ПОСТАНОВКА ПРОБЛЕМИ У ЗАГАЛЬНОМУ ВИГЛЯДІ ТА ЇЇ ЗВ'ЯЗОК ІЗ ВАЖЛИВИМИ НАУКОВИМИ ЧИ ПРАКТИЧНИМИ ЗАВДАННЯМИ

Мультиагентні системи (MAS) є потужним інструментом для виявлення вірусів, адже дозволяють створювати розподілені та інтегровані рішення для виявлення й нейтралізації вірусів, що є особливо важливим при боротьбі з сучасними складними та динамічними загрозами. Така система складається з кількох інтелектуальних агентів (ІА), кожен з яких виконує свою специфічну роль у процесі виявлення і нейтралізації вірусів, її розробка є доцільною для виявлення поліморфних вірусів.

АНАЛІЗ ОСТАННІХ ДОСЛІДЖЕНЬ І ПУБЛІКАЦІЙ

Значна кількість досліджень вчених присвячена різним методам, технікам і підходам до аналізу та виявлення зловмисного програмного забезпечення (ЗПЗ) [1, 2]. Робота [3] відображає комплексний сучасний огляд досліджень підходів до виявлення ЗПЗ. Серед них можна відзначити машинне навчання [4-8], згорткові нейронні мережі (CNN) [9-11], метод, заснований на механізмі уваги [12], динамічний метод виявлення під назвою штучне виявлення зловмисних програм (AMD) [13], метод аналізу головних компонент [14], система на основі емуляції під назвою UBER для покращення пісочниці аналізу ЗПЗ [15], аналіз основних компонент (PCA) використовується для вилучення функцій, тоді як комбінація вихідних кодів з виправленням помилок (ECOC) і адаптивної системи нейронечіткого висновку (ANFIS) використовується для класифікації [16], модель, яка реалізувала класифікатор випадкового лісу для вибору ознак і модель опорних векторних машин (SVM) для виявлення ЗПЗ [17], алгоритми: Boyer-Moore, Aho–Corasick, Naïve String пошуку, Rabin Karp String Search та Knuth–Morris–Pratt [18], архітектура запропонованої системи наближена до традиційної нейро-нечіткої системи Takagi-Sugeno-Kang [19], наскрізна модель для виявлення мережевих атак і класифікації мережевих атак з використанням рекурентних моделей на основі глибокого навчання [20].

Агент – це комп'ютерна система, розміщена в зовнішньому середовищі, здатна взаємодіяти з ним, виконуючи автономні раціональні дії для досягнення цілей [21-23]. Формальну модель інтелектуального агента (ІА) в багатоагентних системах, використовуючи дискретну математику, можна описати як систему окремих агентів, які взаємодіють один з одним. Кожен агент представлений як дискретна сутність із визначеним станом. Стан агента може змінюватися з часом відповідно до набору правил або функцій, які базуються на поточному стані агента та стані його середовища. Ці правила або функції можна виразити за допомогою різних дискретних математичних структур, таких як графіки, послідовності та набори. Наприклад, зв'язки між агентами можна представити у вигляді графа, де кожен вузол представляє агента, а кожне ребро – можливу взаємодію між двома агентами.

Крім того, послідовність дій, які виконує агент, можна представити у вигляді послідовності станів, де кожному стану відповідає дія. Набір усіх можливих дій для агента можна представити як набір, а стратегію агента можна визначити як функцію, яка відображає поточний стан агента на дію в цьому наборі. Ця формальна модель дозволяє аналізувати та прогнозувати поведінку багатоагентних систем за допомогою дискретних математичних методів. Це дозволяє досліджувати різні властивості системи, такі як стабільність, конвергенція та оптимальність.

Мультиагентна система (MAS), по суті, є набором кількох взаємодіючих ІА. Кожен агент, у своїй найпростішій формі, є обчислювальною сутністю, яка відчуває навколишнє середовище та реагує відповідно. Ці агенти здатні до автономної поведінки, можуть вчитися на своєму досвіді та мають здатність взаємодіяти з іншими агентами в системі. Відносини між агентами всередині системи можна представити у вигляді графа. На цьому графіку кожен вузол представляє агента, а кожне ребро представляє потенційну взаємодію між двома агентами. Це візуальне представлення дозволяє чітко зрозуміти та проаналізувати взаємодію всередині системи. Подібним чином ряд дій, які виконує агент, можна представити у вигляді послідовності станів. Кожен стан у цій послідовності відповідає певній дії, яку виконує агент. Цей послідовний підхід забезпечує чітке, покрокове представлення дій агента, що дозволяє детально аналізувати та розуміти.

Як альтернатива, повний діапазон можливих дій, доступних для агента в системі, може бути представлений як набір. Використовуючи цей підхід, стратегію агента можна визначити як функцію, яка відображає поточний стан агента на дію в цьому наборі. Цей метод забезпечує повний огляд усіх потенційних дій, сприяючи глибшому розумінню поведінки агента.

MAS досить часто використовуються для виявлення та аналізу ЗПЗ [24-26]. Також деякі напрацювання у даному напрямку представлено у роботах автора [27, 28].

У роботі [29] розроблено концептуальну модель багатокомп'ютерних систем, яка призначена для забезпечення функціонування антивірусних приманок і пасток для виявлення шкідливих програм і комп'ютерних атак у корпоративних мережах. Виявлення ЗПЗ розподіленими системами з частковою централізацією [30].

Серед антивірусних програм, які активно використовуються для виявлення поліморфних вірусів можна відзначити: Symantec (США), Bitdefender (Румунія), McAfee (США), ESET NOD32 (Словаччина), Trend Micro (Японія), Sophos (Великобританія), Malwarebytes (США), Avast (Чехія), F-Secure (Фінляндія).

Порівняння зазначених антивірусних програм стосовно використання методів виявлення та можливостей аналізу темпів поширення та класифікації поліморфних вірусів відображено у таблиці 1.

Згідно таблиці 1 можна помітити, що жодна із розглянутих антивірусних програм не використовує метод виявлення поліморфних вірусів Probabilistic Logic Network (PLN), а також є прогалини стосовно класифікації за рівнями складності (рівнями поліморфізму). Лише Bitdefender (Румунія) та Trend Micro (Японія) здійснюють цей процес на високому рівні.

Таблиця 1

Порівняння антивірусних програм стосовно використання методів виявлення та можливостей аналізу темпів поширення та класифікації поліморфних вірусів

Антивірусна програма	Аналіз темпів поширення поліморфних вірусів	Методи виявлення поліморфних вірусів						Класифікація за рівнями складності (рівнями поліморфізму)	Час реакції	Виявлення нових загроз
		Евристичний аналіз	Аналіз поведінки	Машинне навчання	Хмарний аналіз	Анти-обфускація	Probabilistic Logic Network (PLN)			
Symantec (США) [31]	Середній	+	+	+	+	+	-	Середній	Середній	Високе
Bitdefender (Румунія) [32]	Високий	+	+	+	+	+	-	Високий	Швидкий	Високе
McAfee (США) [33]	Середній	+	+	+	+	+	-	Середній	Швидкий	Високе
ESET NOD32 (Словаччина) [34]	Середній	+	+	+	-	+	-	Середній	Швидкий	Високе
Trend Micro (Японія) [35]	Високий	+	+	+	-	+	-	Високий	Швидкий	Високе
Sophos (Великобританія) [36]	Середній	+	+	+	-	+	-	Середній	Швидкий	Високе
Malwarebytes (США) [37]	Низький	+	+	+	-	-	-	Низька	Середній	Високе
Avast (Чехія) [38]	Високий	+	+	+	-	+	-	Середній	Швидкий	Високе
F-Secure (Фінляндія) [39]	Середній	+	+	+	-	+	-	Середній	Середній	Високе

ВИДІЛЕННЯ НЕВИРІШЕНИХ РАНІШЕ ЧАСТИН ЗАГАЛЬНОЇ ПРОБЛЕМИ, КОТРИМ ПРИСВЯЧУЄТЬСЯ СТАТТЯ

Хоча науковці присвячують значну кількість своїх досліджень даній проблемі, питання актуальної архітектури MAS для виявлення поліморфних вірусів вивчено недостатньо та вимагає подальших досліджень. Особливо це стосується проблеми дослідження темпу поширення поліморфних вірусів та визначення необхідних методів їх виявлення; використання методу Probabilistic Logic Network (PLN), який надає додаткові можливості для виявлення поліморфних вірусів; здійснення нечіткої класифікації поліморфних вірусів згідно рівнів складності та визначення стратегії прийняття рішення, враховуючи типи загроз у їх комплексному поєднанні.

ФОРМУЛЮВАННЯ ЦІЛЕЙ СТАТТІ

Метою статті є розробка архітектури MAS для виявлення та класифікації поліморфних вірусів, яка каскадно делегуючи повноваження, зможе реалізувати аналіз середовища (перевірку системи на наявність підозрілих або шкідливих дій, аналізуючи файлову систему, оперативну пам'ять, мережевий трафік та поведінку процесів); дослідить темпи поширення поліморфних вірусів та встановить необхідні методи для їх виявлення; використає комплекс методів виявлення поліморфних вірусів та сформує їх «чорний список»; здійснить нечітку класифікацію поліморфних вірусів згідно рівнів їх складності; використає різні стратегії прийняття рішення, враховуючи типи загроз та характер поліморфних вірусів, а також здійснить оновлення баз даних для більш точного виявлення нових варіантів вірусів.

ВИКЛАД ОСНОВНОГО МАТЕРІАЛУ

У попередньому дослідженні автора [27] запропонована модель MAS для виявлення поліморфних вірусів. Дане дослідження пропонує її удосконалену версію, яка інтегрує попередні дослідження автора.

На рисунку 1 представлена архітектура MAS для виявлення поліморфних вірусів. MAS відображається як сукупність агентів $A = \{A_1, A_2, ..., A_n\}$, які взаємодіють із комп'ютерною системою та один з одним.

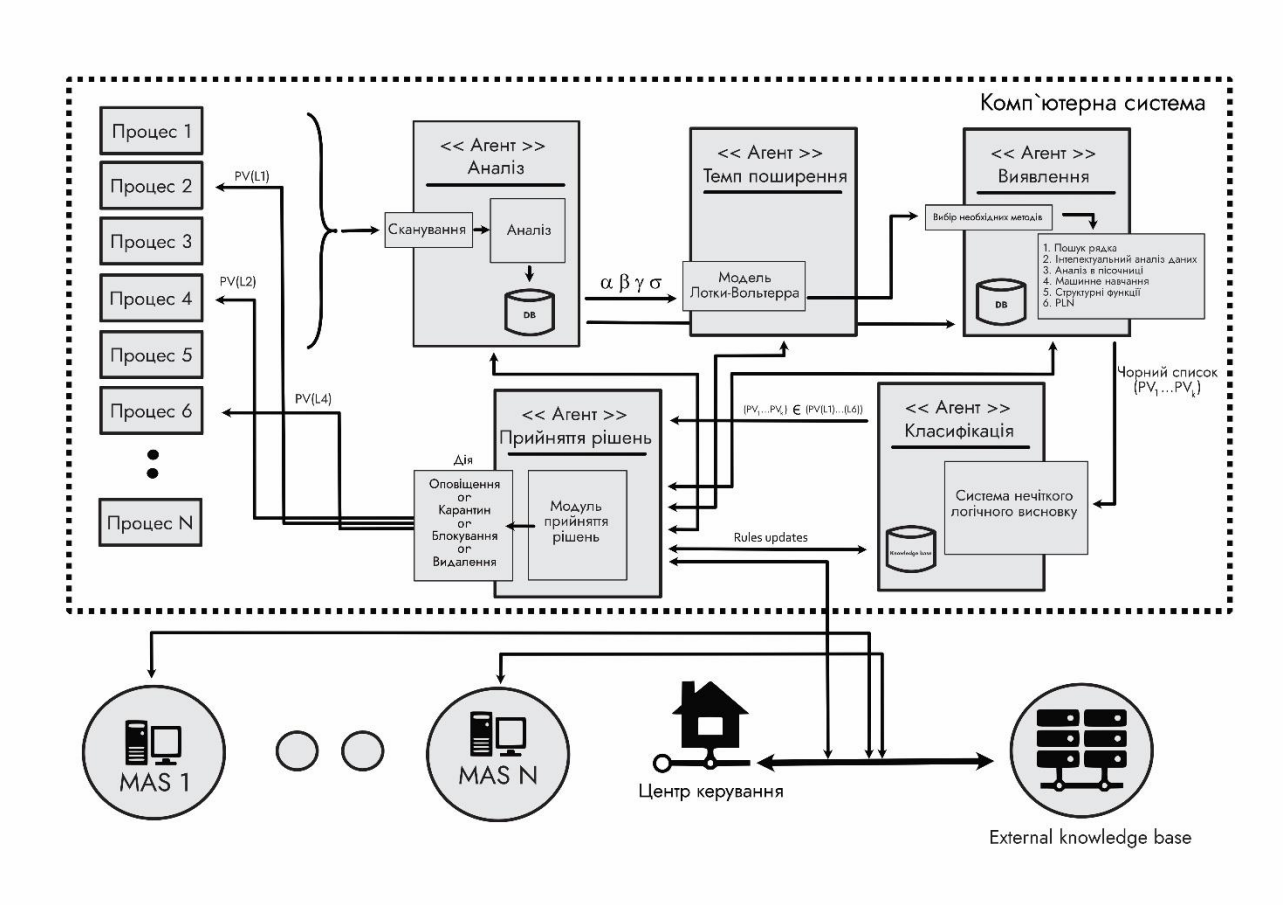

Рис. 1. Архітектура мультиагентної системи виявлення поліморфних вірусів в комп'ютерних системах

Згідно рисунку 1, запропонована MAS складається з наступних ІА: ІА «Аналіз», ІА «Темп поширення», ІА «Виявлення», ІА «Класифікація», ІА «Прийняття рішення», які виконують встановлені ролі та взаємодіють між собою.

ІА «Аналіз» перевіряє систему на наявність підозрілих або шкідливих дій, аналізуючи файлову систему, оперативну пам'ять, мережевий трафік та поведінку процесів; включає збір інформації, що стосується програмного забезпечення, та проведення глибокого аналізу для визначення характеру загрози, її джерела та можливого впливу. В результаті аналізу системи ІА «Аналіз» формує та передає ІА «Темп поширення» значення коефіцієнтів *α, β, γ, δ.*

ІА «Темп поширення» здійснює моделювання процесу поширення поліморфних вірусів на основі моделі Лотки-Вольтерра. Даний агент досліджує темп поширення поліморфних вірусів та визначає, яку кількість та які методи виявлення поліморфних вірусів необхідно використати ІА «Виявлення».

У загальному вигляді модель міжвидової конкуренції виглядає наступним чином:

$$\begin{cases} \frac{dx}{dt} = (\alpha - \beta y)x \\ \frac{dy}{dt} = (-\gamma + \delta x)y \end{cases} \quad (1)$$

де x – кількість жертв (кількісний вимір поліморфних вірусів на момент часу t); y – чисельність хижаків (кількісний вимір доступних технологій, методів та інструментів виявлення поліморфних вірусів на момент часу t); t – час (час виявлення поліморфних вірусів); $\alpha, \beta, \gamma, \delta$ – коефіцієнти, котрі відображають взаємодію між видами.

У випадку адаптації моделі для моделювання процесу виявлення поліморфних вірусів $\alpha, \beta, \gamma, \delta$

відображають наступне: α – ймовірність того, що кількість поліморфних вірусів зростатиме; β – ймовірність того, що поліморфні віруси різних рівнів складності будуть виявлені за допомогою обраних методів, технологій та інструментів; γ – ймовірність того, що деякі з обраних методів, технологій та інструментів не будуть ефективними при виявленні поліморфних вірусів різних рівнів складності в результаті появи нових їх різновидів; δ – ймовірність того, що поліморфні віруси різних рівнів складності вимагатимуть комплексного використання обраних методів, технологій та інструментів, а також новітніх підходів.

ІА «Виявлення» використовує комплекс методів для виявлення поліморфних вірусів (кількість методів визначена ІА «Темп поширення»): алгоритми пошуку рядка, інтелектуальний аналіз даних, аналіз в пісочниці, машинне навчання, метод розробки структурних функцій, PLN. Більш детально даний підхід описано у роботі автора [28].

ІА «Класифікація» здійснює нечітку класифікацію виявлених поліморфних вірусів (ІА «Виявлення») згідно 6 рівнів їх складності.

Поліморфні віруси першого рівня поліморфізму застосовують сталі значення для різних розшифровувачів. Виявляються вони за незмінними ділянками коду розшифровувача.

Формальна модель поліморфних вірусів визначається як сукупність:

$$P_1 = (B, A, D, H, K, \xi, \sigma, \rho, R), \tag{2}$$

де B – сукупність інструкцій користувацької програми чи системного процесу, вірогідно інфікованих поліморфним вірусом, $B = \{b_i \mid i=1,2, \ldots, n\}$; A – сукупність алгоритмів розшифровування присутніх у поліморфному вірусі, $A = \{a_i \mid i=1,2 \ldots, x\}$; D – сукупність інструкцій вірусу та розшифровувача, $D = \{d_i \mid i=1,2 \ldots, y\}$; H – сукупність інструкцій вірусу для вибору одного з наявних у вірусі алгоритмів розшифровування, $H = \{h_i \mid i=1,2, \ldots, j\}$; K – сукупність команд, які є основним кодом вірусу, $K = \{k_i \mid i=1,2 \ldots, v\}$; ξ - функція вибору a_i розшифровувача, $\xi: H \to A$; σ – функція створення зловмисних команд (тіла вірусу) шляхом виконання команд $d_{a_i} \in D$ розшифровувача a_i, $\sigma: D_{x_i} \to K$; ρ – функція створення поведінки поліморфного вірусу R шляхом вкорінення тіла вірусу K в команди програми В, $\rho: B \times K \to R$.

Другий рівень поліморфізму вірусів характеризується розшифровувачами які мають постійну одну або декілька команд по яким їх можна виявити. Виявлення такого типу поліморфних вірусів можливе за специфічною сигнатурою в розшифровувачі.

Формальна модель поліморфних вірусів другого рівня визначається як сукупність:

$$P_2 = (B, L, K, \rho, \varepsilon, R), \tag{3}$$

де B – сукупність інструкцій користувацької програми чи системного процесу, вірогідно інфікованих поліморфним вірусом, B = {bi | i=1,2, …, n}; L – сукупність інструкцій, які складають код розшифровувача, L = {li, | i=1,2, …, x}; K – сукупність команд, які є основним кодом вірусу, K = {ki | i=1,2 …, v}; ρ – функція створення поведінки поліморфного вірусу R шляхом вбудовування тіла вірусу K в програму B, $\rho: B \times K \to R$; ε – функція створення інструкцій основного коду вірусу шляхом вибору команд розшифровувача, $\varepsilon: L \to K$.

Поліморфні віруси, що використовують в розшифровувачі інструкції, які не мають ролі в розшифруванні вірусного коду, чи «команди-сміття», відносять до третього рівня поліморфізму. Ці віруси, також, можна виявити за допомогою сталої сигнатури, якщо провести фільтрацію та видалення команд-сміття.

Поліморфні віруси четвертого рівня використовують в розшифровувачі взаємозамінювані інструкції без зміни алгоритму розшифровування.

Формальна математична модель поліморфних вірусів третього та четвертого рівнів визначається сукупністю:

$$P_{3,4} = (B, L, K, T, \varphi, \phi, R), \tag{4}$$

де B – сукупність інструкцій користувацької програми чи системного процесу, вірогідно інфікованих поліморфним вірусом, $B = \{b_i \mid i=1,2, \ldots, n\}$; L – сукупність інструкцій, які складають код розшифровувача, $L = \{l_i, \mid i=1,2, \ldots, x\}$; K – сукупність команд, які є основним кодом вірусу, $K = \{k_i \mid i=1,2 \ldots, v\}$; T – множина інструкцій, які називаються «команди-сміття», $T = \{t_i \mid i = 1,2, \ldots, y\}$; φ – функція створення основного коду вірусу засобами розшифровувача, який вбудовує «команди-сміття» в сукупність зловмисних команд, $\varphi: L \times T \to K$; ϕ – функція створення поведінки поліморфного вірусу R шляхом вбудовування тіла вірусу K в користувацьку програму B, $\phi = B \times K \to R$.

Формальна математична модель поліморфних вірусів п’ятого рівня визначається сукупністю:

$$M_5 = (B, A, H, K, T, G, \xi, \theta, \eta, R), \tag{5}$$

де B – сукупність інструкцій користувацької програми чи системного процесу, вірогідно інфікованих поліморфним вірусом, $B = \{b_i \mid i=1,2, \ldots, n\}$; A – сукупність алгоритмів розшифровування присутніх у поліморфному вірусі, $A = \{a_i \mid i=1,2 \ldots, x\}$; H – сукупність інструкцій вірусу для вибору одного з наявних у вірусі алгоритмів розшифровування, $H = \{h_i \mid i=1,2, \ldots, j\}$; K – сукупність команд, які є основним кодом вірусу, $K = \{k_i \mid i=1,2 \ldots, v\}$; T – множина інструкцій, які називаються «команди-сміття», $T = \{t_i \mid i = 1,2, \ldots, y\}$; G – сукупність інструкцій вірусу a_i та розшифровувача, $G = \{g_i | i = 1,2, \ldots, j\}$; ξ - функція вибору a_i розшифровувача, $\xi: H \to A$; θ – функція створення зловмисних інструкцій засобами вибору a_i розшифровувача інструкціями $g_{x_i} \in G$ і створення чіткого порядку їх виконання, $\theta: T \in G_{x_i} \to K$; η – функція створення поведінки поліморфного вірусу R шляхом вбудовування зловмисних команд K в команди програми B, $\eta: B \times K \to R$.

Шостий рівень поліморфізму вірусів характеризується нешифрованими частинами, що складаються з підпрограм, які «перемішуються» всередині тіла вірусу. Такі віруси називаються - пермутуючі віруси.

Формальна модель поліморфних вірусів шостого рівня визначається сукупністю:

$$P_6 = (B, L, K, \vartheta, R), \tag{6}$$

де B – сукупність інструкцій користувацької програми чи системного процесу, вірогідно інфікованих поліморфним вірусом, $B = \{b_i \mid i=1,2, \ldots, n\}$; L – сукупність інструкцій, які складають код розшифровувача, $L = \{l_i, \mid i=1,2, \ldots, x\}$; K – сукупність команд, які є основним кодом вірусу, $K = \{k_i \mid i=1,2 \ldots, v\}$; ϑ – функція створення поведінки поліморфного вірусу R шляхом розміщення інструкцій коду, інструкцій розшифровування, інструкцій тіла вірусу блоками за заданим алгоритмом порядком, $\vartheta: B \times L \times K \to R$.

Нечітка класифікація здійснюється за алгоритмом: визначення характеристик виявлених поліморфних вірусів та формування дерева логічного висновку; опис лінгвістичних змінних; визначення функцій належності лінгвістичних термів; формування бази знань системи нечіткого висновку; визначення ймовірності належності досліджуваного файлу до поліморфних вірусів різних рівнів складності; нечітка класифікація поліморфних вірусів.

Були встановлені наступні показники для аналізу виявлених поліморфних вірусів: рівень шифрування коду (*CE*), рівень обфускації коду (*CO*), рівень використання самозмінюваного та динамічного коду (*SDC*), рівень захисту від аналізу та обходу пісочниць (*PAS*), рівень складності мережевої поведінки та взаємодії (*CNB*), рівень взаємодії із системою та реєстром (*ISR*), рівень адаптивної поведінки та самозахисту (*ABS*).

Використано наступне співвідношення для визначення рівня складності поліморфного вірусу (LPV):

$$LPV = f_{LPV}(CE, CO, SDC, PAS, CNB, ISR, ABS). \tag{7}$$

FIS-структура співвідношення 7 у середовищі Matlab відображена на рис. 2.

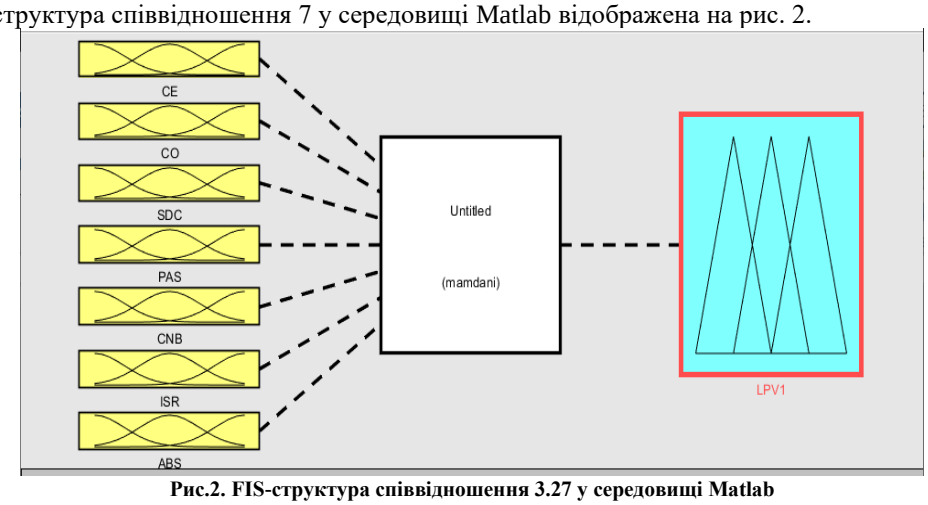

Рис.2. FIS-структура співвідношення 3.27 у середовищі Matlab

Опис лінгвістичних змінних представлено в таблиці 2.

Таблиця 2

Опис лінгвістичних змінних

Параметр	**Назва лінгвістичної змінної (x)**	**Універсальна множина (U)**	**Лінгвістичні терми (T)**
CE	Рівень шифрування коду	(0-6) балів	Level 1 (L1), Level 2 (L2), Level 3 (L3), Level 4 (L4), Level 5 (L5), Level 6 (L6)
CO	Рівень обфускації коду		
SDC	Рівень використання самозмінюваного та динамічного коду		
PAS	Рівень захисту від аналізу та обходу пісочниць		
CNB	Рівень складності мережевої поведінки та взаємодії		
ISR	Рівень взаємодії із системою та реєстром		
ABS	Рівень адаптивної поведінки та самозахисту		
LPV	Ймовірність належності вірусу до досліджуваного рівня складності поліморфних вірусів	(0-100) %	Low (L), Low Medium (LM), Medium (M), High Medium (HM), High (H)

Сформовано базу знань системи нечіткого висновку. Нечіткі логічні рівняння для бази знань для показника *LPV* (1 рівень складності) мають вигляд (фрагмент):

$$\mu^{H}(LPV) = \mu^{L1}(CE) \wedge \mu^{L1}(CO) \wedge \mu^{L1}(SDC) \wedge \mu^{L1}(PAS) \wedge \mu^{L1}(CNB) \wedge \mu^{L1}(ISR) \wedge \mu^{L1}(ABS) \vee \mu^{L2}(CE) \wedge \mu^{L1}(CO) \wedge \mu^{L1}(SDC) \wedge \mu^{L1}(PAS) \wedge \mu^{L1}(CNB) \wedge \mu^{L1}(ISR) \wedge \mu^{L1}(ABS) \vee \mu^{L1}(CE) \wedge \mu^{L2}(CO) \wedge \mu^{L1}(SDC) \wedge \mu^{L1}(PAS) \wedge \mu^{L1}(CNB) \wedge \mu^{L1}(ISR) \wedge \mu^{L1}(ABS) \vee \mu^{L1}(CE) \wedge \mu^{L1}(CO) \wedge \mu^{L2}(SDC) \wedge \mu^{L1}(PAS) \wedge \mu^{L1}(CNB) \wedge \mu^{L1}(ISR) \wedge \mu^{L1}(ABS) \vee \mu^{L1}(CE) \wedge \mu^{L1}(CO) \wedge \mu^{L1}(SDC) \wedge \mu^{L2}(PAS) \wedge \mu^{L1}(CNB) \wedge \mu^{L1}(ISR) \wedge \mu^{L1}(ABS) \vee \mu^{L1}(CE) \wedge \mu^{L1}(CO) \wedge \mu^{L1}(SDC) \wedge \mu^{L1}(PAS) \wedge \mu^{L2}(CNB) \wedge \mu^{L1}(ISR) \wedge \mu^{L1}(ABS) \vee \mu^{L1}(CE) \wedge \mu^{L1}(CO) \wedge \mu^{L1}(SDC) \wedge \mu^{L1}(PAS) \wedge \mu^{L1}(CNB) \wedge \mu^{L2}(ISR) \wedge \mu^{L1}(ABS) \vee \mu^{L1}(CE) \wedge \mu^{L1}(CO) \wedge \mu^{L1}(SDC) \wedge \mu^{L1}(PAS) \wedge \mu^{L1}(CNB) \wedge \mu^{L1}(ISR) \wedge \mu^{L2}(ABS); \quad (7)$$

де μ – функція належності; *H* – лінгвістичний терм *High (H)*; *L1* – лінгвістичний терм *Level 1 (L1); L2* – лінгвістичний терм *Level 2 (L2);* $\vee(\wedge)$ – операція максимуму (мінімуму).

В результаті використання нечіткого логічного висновку для кожного файлу, що виконується, було отримане числове значення (у %) ймовірності належності до кожного з рівнів складності поліморфних вірусів.

ІА «Прийняття рішення» використовує різні стратегії прийняття рішення (оповіщення, блокування, карантин, видалення), враховуючи типи загроз.

Для розрахунку інтегрального показника ефективності MAS пропонується використати наступний підхід: розгляд його в адитивній (E_A) та мультиплікативній формі (E_M).

В адитивній формі:

$$E_A = \sum_{i=1}^{n} a_i E_i^A \,, \quad (8)$$

В мультиплікативній формі :

$$E_M = \prod_{i=1}^{m} (E_i^M)^{a_i} \,, \quad (9)$$

де E_i^A – складові індикатори ефективності MAS для адитивної форми; E_i^M – складові індикатори ефективності MAS для мультиплікативної форми; n – кількість складових для визначення ефективності MAS; a_i – вагові коефіцієнти. При чому:

$$\sum_{i=1}^{n} a_i = 1,\ a_i \geq 0, i = \overline{1,n} \quad (10)$$

Розрахунок ефективності MAS пропонується визначати як усереднене значення інтегрального показника ефективності MAS в адитивній та мультиплікативній формі.

ВИСНОВКИ З ДАНОГО ДОСЛІДЖЕННЯ І ПЕРСПЕКТИВИ ПОДАЛЬШИХ РОЗВІДОК У ДАНОМУ НАПРЯМІ

У роботі вперше розроблено архітектуру MAS виявлення поліморфних вірусів, яка каскадно делегуючи повноваження, здійснює аналіз середовища, досліджує темпи поширення, виявляє, класифікує поліморфні віруси та використовує різні стратегії прийняття рішення, враховуючи типи загроз. В структуру даної системи входять інтелектуальні агенти (ІА): ІА «Аналіз», ІА «Темп поширення», ІА «Виявлення», ІА «Класифікація», ІА «Прийняття рішення», які виконують встановлені ролі та взаємодіють між собою.

Подальші дослідження будуть спрямовані на оптимізацію комунікацій між агентами MAS виявлення поліморфних вірусів (зменшення часу реагування на нові загрози за допомогою більш швидкої та ефективної взаємодії агентів), інтеграцію з іншими системами безпеки, управління конфіденційністю та анонімністю.

References

1. Alenezi M. N., Alabdulrazzaq H., Alshaher A. A., Alkharang M. M. Evolution of Malware Threats and Techniques: A Review. *International Journal of Communication Networks and Information Security (IJCNIS)*. 2020. Vol. 12(3). P. 326–337. DOI: https://doi.org/10.17762/ijcnis.v12i3.4723

2. Nasir M.H., Khan S.A., Khan M.M., Fatima M.. Swarm Intelligence Inspired Intrusion Detection Systems - A Systematic Literature Review. *Computer Networks*. 2022. Vol. 205(1). 108708. DOI: https://doi.org/10.1016/j.comnet.2021.108708

3. Aboaoja F. A., Zainal A., Ghaleb F. A., Al-rimy B. A. S., Eisa T. A. E., Elnour A. A. H. Malware Detection Issues, Challenges, and Future Directions: A Survey. *Applied Sciences*. 2022. Vol. 12(17). DOI: https://doi.org/10.3390/app12178482

4. Lysenko S, Bobrovnikova K, Kharchenko V, Savenko O. IoT Multi-Vector Cyberattack Detection Based on Machine Learning Algorithms: Traffic Features Analysis, Experiments, and Efficiency. *Algorithms*. 2022. Vol. 15(7):239. DOI: https://doi.org/10.3390/a15070239

5. Akhtar M. S., Feng T. Malware Analysis and Detection Using Machine Learning Algorithms. *Symmetry (Basel)*. 2022. Vol. 14(11). P. 2304. DOI:https://doi.org/10.3390/sym14112304

6. Chiwariro R., Pullagura L. Malware Detection and Classification Using Machine Learning Algorithms, International Journal for Research in Applied Science & Engineering Technology. *IJRASET*. 2023. *Vol*. 11. P. 1727-1738. DOI: 10.22214/ijraset.2023.55255

7. Kurian A. J., Santhosh A., Subin M. Enhanced malware detection framework leveraging machine learning algorithms. *International Research Journal of Modernization in Engineering Technology and Science.* 2024. Vol. 06(03). P. 3597-3603.

8. Salem A., Banescu S., Pretschner V. Maat: Automatically Analyzing VirusTotal for Accurate Labeling and Effective Malware Detection. *ACM Transactions on Privacy and Security*. 2021. Vol. 24. P. 1–35. DOI: https://doi.org/10.48550/arXiv.2007.00510

9. Chakraborty A., Kriti K., Yateendra, Bennet Praba M.S. Polymorphic Malware Detection by Image Conversion Technique. *International Journal of Engineering and Advanced Technology (IJEAT)*. 2020. Vol. 9 (3). P. 2898-2903. DOI: https://doi.org/10.35940/ijeat.B4999.029320

10. Naeem H., Alshammari B. M., Ullah F. Explainable artificial intelligence-based IoT device malware detection mechanism using image visualization and fine-tuned CNN-based transfer learning model. *Computational Intelligence and Neuroscience.* 2022. 7671967. DOI: https://doi.org/10.1155/2022/7671967

11. Nicheporuk A., Savenko O., Nicheporuk A., Nicheporuk Y. An android malware detection method based on CNN mixed-data model. *CEUR Workshop Proceedings*. 2020. Vol. 2732. P.198–213.

12. Choi S., Bae J., Lee C., Kim Y., Kim J. Attention-based automated feature extraction for malware analysis. *Sensors (Switzerland).* 2020. Vol. 20(10). P. 1–17. DOI: https://doi.org/10.3390/s20102893

13. Jerbi M., Dagdia Z. C., Bechikh S., Said L. B. On the Use of Artificial Malicious Patterns for Android Malware Detection. *Computer & Security*. 2020. Vol. 92. 101743. DOI: https://doi.org/10.1016/j.cose.2020.101743

14. Komar M., Golovko V., Sachenko A., Bezobrazov S. Intelligent system for detection of networking intrusion. *In Proc. of the 6th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems*, Prague, Czech Republic, 2011, pp. 374-377. DOI: https://doi.org/10.1109/IDAACS.2011.6072777

15. Liu S., Feng P., Wang S., Sun K., Cao J. Enhancing malware analysis sandboxes with emulated user behavior. *Computers and Security*. 2022. Vol. 115. 102613. DOI: https://doi.org/10.1016/j.cose.2022.102613

16. Majidian Z., TaghipourEivazi S., Arasteh B., Babai S. An Intrusion Detection Method to Detect Denial of Service Attacks Using Error-Correcting Output Codes and Adaptive Neuro-Fuzzy Inference. *Computers and Electrical Engineering*, 2023. Vol. 106. 108600. DOI: https://doi.org/10.1016/j.compeleceng.2023.108600

17. Nwagwu C. B., Taylor O. E., Nwiabu N. D. A Model for Detection of Malwares on Edge Devices. *International Journal Of Engineering And Computer Science*. 2024. Vol. 13(07). P. 26274-26283. DOI: https://doi.org/10.18535/ijecs/v13i07.4846

18. Obeidat I., AlZubi M. Developing a Faster Pattern Matching Algorithms for Intrusion Detection System. *International Journal of Computing*. 2019. Vol. 18(3). P. 278-284. DOI: https://doi.org/10.47839/ijc.18.3.1520

19. Perova I., Bodyanskiy Y. Fast medical diagnostics using autoassociative neuro-fuzzy memory. *International Journal of Computing*. 2017. Vol. 16(1). P. 34-40. DOI: https://doi.org/10.47839/ijc.16.1.869

20. Ravi V., Chaganti R., Alazab M. Recurrent Deep Learning-Based Feature Fusion Ensemble Meta-Classifier Approach for Intelligent Network Intrusion Detection System. *Computers & Electrical Engineering*. 2022. Vol. 102(3). 108156. DOI: https://doi.org/10.1016/j.compeleceng.2022.108156

21. Abu Bakar R., Huang X., Javed M.S., Hussain S., Majeed M.F. An Intelligent Agent-Based Detection System for DDoS Attacks Using Automatic Feature Extraction and Selection. *Sensors*. 2023. Vol. 23. 3333. DOI: https://doi.org/10.3390/s23063333

22. Ligo A.K., Kott A., Linkov I. How to measure cyber-resilience of a system with autonomous agents: approaches and challenges. *IEEE Engineering Management Review*. 2021. Vol. 49(2). P. 89-97. DOI: https://doi.org/10.1109/EMR.2021.3074288

23. Taher F., AlFandi O., Al-kfairy M., Al Hamadi H., Alrabaee S. DroidDetectMW: A Hybrid Intelligent Model for Android Malware Detection. *Applied Sciences*. 2023. Vol. 13. 7720. DOI: https://doi.org/10.3390/app13137720

24. Abidar R., Moummadi K., Moutaouakkil F., Medromi H. Intelligent and Pervasive Supervising Platform for Information System Security Based on Multi-Agent Systems. *International Review on Computers and Software (IRECOS)*. 2015. Vol. 10 (1). DOI: https://doi.org/10.15866/irecos.v10i1.4699

25. Brahmi I., Ben Yahia S., Aouadi H., Poncelet P. (2012). Towards a Multiagent-Based Distributed Intrusion Detection System Using Data Mining Approaches. *Lecture Notes in Computer Science*. 2012. Vol. 7103. DOI: https://doi.org/10.1007/978-3-642-27609-5_12

26. Monga R., Karlapalem K. MASFMMS: Multi Agent Systems Framework for Malware Modeling and Simulation. *Lecture Notes in Computer Science*. 2009. Vol. 5269. DOI: https://doi.org/10.1007/978-3-642-01991-3_8

27. Чайковський М. Модель мультиагентної системи для виявлення поліморфних вірусів. *Herald of Khmelnytskyi National University. Technical Sciences*. 2024. Vol. *347*(1). С.543-547. DOI: https://doi.org/10.31891/2307-5732-2025-347-74

28. Чайковський М. Комплексний підхід до виявлення та аналізу поліморфного зловмисного програмного забезпечення. *Міжнародний науково-технічний журнал «Вимірювальна та обчислювальна техніка в технологічних процесах».* 2024. № 2. С. 42-50. URL: https://doi.org/10.31891/2219-9365-2024-78-5

29. Kashtalian A., Lysenko S., Savenko,O., Nicheporuk,A., Sochor T., Avsiyevych V. Multi-computer malware detection systems with metamorphic functionality. *Radioelectronic and Computer Systems*. 2024. Vol. 1. P. 152-175. DOI: https://doi.org/10.32620/reks.2024.1.13

30. Savenko B., Kashtalian A., Lysenko S., Savenko O. Malware Detection By Distributed Systems with Partial Centralization. 2023 IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Dortmund, Germany, 2023, pp. 265-270, DOI: 10.1109/IDAACS58523.2023.10348773
31. Symantec Endpoint Security Complete. Available at: https://www.broadcom.com/products/cybersecurity/endpoint/end-user/complete (accessed 21.02.2025).
32. Bitdefender Antivirus Plus. Available at: https://bitdefender.ua/ (accessed 21.02.2025).
33. McAfee Antivirus. Available at: https://www.mcafee.com/en-gb/index.html (accessed 21.02.2025).
34. ESET NOD32 Antivirus. Available at: https://www.eset.com/ua/ (accessed 21.02.2025).
35. Trend Micro. Available at: https://www.trendmicro.com/en_gb/business.html (accessed 21.02.2025).
36. Sophos. Available at: https://home.sophos.com/en-us (accessed 21.02.2025).
37. Malwarebytes. Available at: https://www.malwarebytes.com/ (accessed 21.02.2025).
38. Avast. Available at: https://www.avast.ua/ (accessed 21.02.2025).
39. F-Secure. Available at: https://www.f-secure.com/en (accessed 21.02.2025).