https://doi.org/10.31891/2219-9365-2025-84-16
УДК 005.21:005.8:004.8.

БОХОНЬКО Олександр
Хмельницький національний університет
https://orcid.org/0000-0002-7228-9195
ЛИСЕНКО Сергій
Хмельницький національний університет
https://orcid.org/0000-0001-7243-8747

МЕТОД СИНТЕЗУ РОЗПОДІЛЕНОЇ КОМП'ЮТЕРНОЇ СИСТЕМИ, СТІЙКОЇ ДО АТАК СОЦІАЛЬНОЇ ІНЖЕНЕРІЇ

У статті представлено метод синтезу архітектури системи розподіленої IT – інфраструктури, стійкої до атак соціальної інженерії. Основна ідея полягає в побудові ієрархічної багатоагентної моделі, у якій агент прийняття рішень верхнього рівня координує роботу сервісних агентів, спеціалізованих за модальностями та відповідальних за збирання й перевірку ознак на основі текстових, структурних і контекстних сигналів. Архітектура інтегрує кілька ключових компонентів: модуль агрегації індикаторів, що накопичує характеристики стану в часі; механізм моделювання, який відтворює поведінку користувача та зловмисника для навчання в умовах невизначеності; а також менеджер взаємодій, що планує запити та інформаційні діалоги, використовуючи апріорні знання графа, який кодує ймовірність появи певних індикаторів за наявності конкретного типу атаки.

Навчання системи керується підкріплювальними сигналами, що винагороджують зменшення прогнозної невизначеності (ентропії) та штрафують повторювані або малозначущі дії, що забезпечує формування коротких, інформативних послідовностей взаємодії та каліброваних рішень. Запропонований метод реалізовано на об'єднаному корпусі електронних листів і вебсторінок, анотованих за розширеною таксономією тактик соціальної інженерії, і показано, що ієрархічний дизайн суттєво покращує як бінарне виявлення атак, так і детальну класифікацію тактик / типів порівняно з одноагентними та іншими неієрархічними базовими моделями.

В експериментах запропонована система досягає вищої точності за меншої кількості дій на один приклад, знижує частку хибнопозитивних спрацювань у строгих режимах роботи та демонструє кращу переносимість між модальностями завдяки узгодженій координації спеціалізованих агентів. Метод забезпечує формалізований метод до інженерії багаторівневих адаптивних механізмів захисту від еволюційних атак соціальної інженерії в реальних експлуатаційних умовах.

Ключові слова: ієрархічна багатоагентна система; навчання з підкріпленням; розподілена комп'ютерних систем; соціальна інженерія; мультимодальний аналіз; граф знань; ентропійна винагорода; детектори ознак.

BOKHONKO Oleksandr, LYSENKO Sergii
Khmelnytskyi National University

METHOD FOR SYNTHESIZING A DISTRIBUTED COMPUTER SYSTEM RESILIENT TO SOCIAL ENGINEERING ATTACKS

The paper presents a method for synthesising system architecture for distributed IT – infrastructure systems resistant to social engineering attacks. The core idea is a hierarchical multi-agent design in which a high-level decision-making agent coordinates modality-specialized service agents that collect and verify evidence across textual, structural, and contextual signals.

The architecture integrates several components that aggregates indicators over time, a simulation engine that models user and adversary behavior for training under uncertainty, and an interaction manager that plans information-gathering dialogs with knowledge-graph priors that encode the likelihood of observing particular indicators given an attack type. Learning is guided by reinforcement signals that reward reductions in predictive uncertainty (entropy) and penalize repeated or low-information actions, enabling short, informative interaction sequences and calibrated decisions.

The approach was instantiated on a combined corpus of emails and web pages labeled with a comprehensive taxonomy of social-engineering tactics and demonstrated that the hierarchical design improves both binary detection and fine-grained tactic/type classification compared with single-agent and other non-hierarchical baselines. In the experiments the proposed system achieves higher accuracy while requiring fewer actions per example, lowers the false-positive rate at stringent operating points, and exhibits improved transfer across modalities through principled coordination of specialized agents. The method provides a principled path to engineering multi-level, adaptable defenses against evolving social-engineering threats in operational environments.

Keywords: hierarchical multi-agent system; reinforcement learning; distributed IT infrastructure; social engineering; multimodal analysis; knowledge graph; entropy-based reward; feature detectors.

Стаття надійшла до редакції / Received 30.09.2025
Прийнята до друку / Accepted 22.08.2025

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

Атаки соціальної інженерії залишаються одним із найскладніших класів загроз для сучасних розподілених комп'ютерних систем. На відміну від технічних векторів впливу, соціально-інженерні атаки використовують людські когнітивні вразливості, багатоканальність взаємодій та змінювані тактики, що ускладнює застосування традиційних сигнатурних або статичних моделей захисту.

За останні роки кількість фішингових повідомлень, клонування профілів та комбінованих сценаріїв «email - web - OSN» продовжує стабільно зростати, що підтверджують результати систематичних оглядів сучасних методів машинного навчання й глибинних моделей для виявлення фішингу [2], [3]. Сучасні підходи використовують переважно окремі модальності — електронні листи, URL-адреси, DOM-структури вебсторінок або поведінкові патерни користувачів [1], [4], [5].

Попри успішність таких моделей у локальних сценаріях, вони мають два фундаментальні недоліки. По-перше, більшість рішень обмежені централізованими точками прийняття рішень, а отже не враховують гетерогенність і динаміку комп'ютерної системи. По-друге, ці системи здебільшого працюють без контекстної координації між каналами взаємодії, що робить їх менш ефективними проти складних багатокрокових маніпулятивних сценаріїв.

Розвиток мультимодальних моделей, у яких поєднано текстові, візуальні, поведінкові та структурні ознаки, зокрема EM-BERT-моделі, LSTM-архітектури або гібридні CNN/MLP-детектори [4], [5], демонструє істотний прогрес у точності класифікації атак. Додатковий прорив відбувся завдяки інтеграції модулів пояснюваного ШІ та графів знань, коли залежності між ознаками й класами атак моделюються на рівні семантичних структур [5], [7], [6]. Проте навіть такі системи не враховують розподілену природу взаємодій у корпоративних мережах, де кожен вузол має власний набір ризиків, ознак та можливостей реагування.

Натомість останні роботи демонструють ефективність навчання з підкріпленням (RL) та багатоагентних моделей у задачах реагування на інциденти й оптимізації дій у складних середовищах [8]. Особливо перспективним виявився напрям ієрархічного багатоагентного навчання з підкріпленнням (multi-agent reinforcement learning, H-MARL), у якому менеджер високого рівня координує спеціалізовані підполітики або підлеглі агенти [9], [10].

Такі підходи вже показали покращення збіжності, зменшення кількості помилкових дій та підвищення ефективності реагування. Водночас наявні H-MARL – рішення орієнтовані здебільшого на сценарії мережевого проникнення й incident-response, але не адаптовані до завдань детектування соціальної інженерії, де ключовими є робота з ознаками інформаційного середовища, аналіз поведінки користувачів і комбінування модальностей.

Таким чином, на сьогодні не існує методів, які б одночасно: моделювали комп'ютерну систему як популяцію детекторів з локальними станами; поєднували мультимодальні ознаки різних каналів соціальної інженерії; використовували ієрархічну багатоагентну архітектуру з координацією між вузлами; впроваджували ентропійні й внутрішні винагороди, що знижують інформаційну невизначеність; опиралися на графи знань для пріорів ознак і структурних залежностей.

У цьому контексті пропонується новий метод синтезу архітектури розподіленої КС, стійкої до атак соціальної інженерії, заснований на ієрархічній мультиагентній моделі з елементами середньопольового опису, мультимодальним аналізом ознак та інтегрованою системою винагород. Розроблений метод узгоджує локальні рішення вузлів із глобальною політикою, знижує ентропію стану, підвищує точність класифікації та скорочує кількість дій, необхідних для ідентифікації сценарію атаки, що робить його перспективним для практичного застосування в масштабних корпоративних мережах.

ФОРМУЛЮВАННЯ ЦІЛЕЙ СТАТТІ

Метою дослідження є розроблення та формальне обґрунтування методу синтезу розподіленої комп'ютерної системи, стійкої до атак соціальної інженерії, на основі ієрархічної багатоагентної моделі з використанням мультимодального аналізу ознак, графів знань та механізмів навчання з підкріпленням, який забезпечує узгоджене зменшення невизначеності стану, підвищення точності класифікації та скорочення кількості дій під час виявлення складних багатокрокових сценаріїв соціальної інженерії в розподіленому інформаційному середовищі.

ВИКЛАД ОСНОВНОГО МАТЕРІАЛУ

Огляд сучасних методів виявлення атак соціальної інженерії у розподілених КС

Розглянемо відомі методи виявлення атак соціальної інженерії у розподілених КС.

У роботі [11] запропоновано агентно-модульну архітектуру PhishDebate, у якій автономні агенти аналізують URL, HTML, семантичний контент та сигнали імітації бренду, а їхні висновки узгоджуються через «дебатний» механізм за участю ролей Модератора та Судді. Такий метод зменшує ризик модально-специфічних помилок і досягає істиннопозитивної частки 98.2%.

У роботі [12] автори розробляють багатокомпонентну багатоагентну модель для аналізу фішингових електронних листів. У статті пропонується використання п'яти кооперативних агентів (аналіз тексту, URL, метаданих, пояснювальна складова та адверсарний агент), а їхній внесок зважується за допомогою proximal policy optimization (PPO). Важливою інновацією є адверсарний агент, який генерує перефразовані та модифіковані версії фішингових повідомлень, що підвищує адаптивність системи до нових стилістичних варіацій. Робота демонструє точність 97.89% та стійкість до URL-скорочувачів, підвищуючи надійність мультимодального аналізу.

У статті [13] три окремі моделі обробляють URL-рядки, контент вебсторінки та DOM-структурні особливості, після чого формується остаточна класифікація на основі найвищого рівня впевненості. Такий паралельний аналіз знижує ймовірність корельованих помилок і забезпечує точність 99.21% у поєднанні з низьким рівнем хибнопозитивних спрацювань.

У роботі [14] запропоновано використання мультимодального агентного підходу, який поєднує багатомовні великі мовні моделі з online- і offline-джерелами знань. Запити формуються з візуальних (логотипи, фавікони) та структурних (HTML/DOM) сигналів, а застосування top-k retrieval із зовнішніх баз знань дозволяє підвищити повноту та знизити FP/FN, особливо для рідкісних брендів.

У статті [15] піднімається питання практичного розгортання систем виявлення на стороні користувача. Пропонується інтеграція браузерного розширення, Docker-контейнерів та BiLSTM-моделі з attention-механізмом для виконання класифікації URL у реальному часі. Оцінено три стратегії агрегування ознак (SPhS, MSS, WeAS), з яких WeAS показує найкраще співвідношення точності та швидкодії.

У роботі [16] розглядається виявлення клонування профілів в онлайн-соціальних мережах шляхом об'єднання семантичних атрибутів профілю, стилометричних характеристик тексту та реляційно-графових ознак. Запропонований ансамблевий метод досягає точності близько 95% та забезпечує стійкість за умов неповних або зашумлених даних.

У статті [17] представлено децентралізовану багатоагентну модель для онлайн-виявлення мережевих вторгнень, яка поєднує локальні оновлення моделей із глобальним агрегуванням у межах федеративного навчання. Демонструється, що CNN досягають точності понад 95%, а LSTM можуть виявляти атаки в межах перших 15 пакетів трафіку.

У роботі [18] надано огляд використання федеративного навчання для систем виявлення вторгнень у різних доменах — від IoT до корпоративних мереж — із визначенням ключових викликів, таких як отруєння моделей і забезпечення безпечної агрегації градієнтів.

У статті [19] запропоновано ієрархічну архітектуру PRISM для багатостадійного виявлення вторгнень, де поведінкове семплювання трафіку поєднується з прихованими марковськими моделями (HMM), що приводить до прискорення обробки в 7.5 разів порівняно з централізованими системами.

У роботі [20] представлено гібридний конвеєр, який поєднує min–max нормалізацію, Equilibrium Optimizer та LSTM-AE модель, оптимізовану Snow Ablation Optimizer, досягаючи точності 99.46% на наборі CIC-IDS2017.

У статті [21] описано дворівневий фільтраційний механізм, де первинний частинковий фільтр усуває очевидні аномалії, а вторинний – фокусується на прихованих відхиленнях, зменшуючи хибні тривоги у великих потоках трафіку.

У роботі [22] надано комплексний огляд IDS-систем із класифікацією на моделі зловживань та аномалій, аналізом архітектур (централізованих, розподілених, edge-орієнтованих) і визначенням перспектив розвитку, включно з багатоагентними ієрархіями та приватними FL-протоколами.

Для реалізації мультимодальних детекторів у вебсередовищі важливу технічну роль відіграють інструменти попереднього оброблення. Одним із найбільш поширених рішень є бібліотека BeautifulSoup4 [23], яка забезпечує стабільне парсування HTML/DOM і активно використовується в агентних фреймворках [11–15].

У роботах [24] та [25] демонструється ефективність застосування глибинних Q-мереж (DQN) у завданнях керування складними динамічними системами, включно з енергетичними та cloud–edge архітектурами. Це підкреслює універсальність RL-підходів і мотивує їх використання в багатоагентних системах для редукції ентропії стану та оптимізації вибору дій.

У статті [26] проведено огляд методів візуальної аналітики для побудови людино- та машинно-орієнтованих моделей. Включення таких підходів є важливим для інтерпретації рішень багатоагентних систем та підвищення прозорості поведінки агентів.

Основи методу синтезу розподіленої комп'ютерної системи, стійкої до атак соціальної інженерії, на основі ієрархічної багатоагентної системи

У дослідженні пропонується метод синтезу розподіленої комп'ютерної системи, стійкої до атак соціальної інженерії на основі ієрархічної багатоагентної системи з використанням навчання з підкріпленням. Метод дозволяє побудувати розподілену комп'ютерну систему, здатну виявляти атаки соціальної інженерії, та спирається на взаємодію спеціалізованих компонентів, що відображають ключові процеси аналізу інформаційного середовища.

В основі лежить агент, який, у координації з класифікатором атак, формулює політику діалогу між системою, користувачем та іншими джерелами загрози. Агент керує збором контекстних та технічних характеристик повідомлення або взаємодії, враховуючи вхідну інформацію, яка може свідчити про атаку соціальної інженерії, наприклад, лексико – семантичні особливості тексту електронного листа, параметри мережевого з'єднання, поведінкові характеристики користувача в каналі зв'язку тощо. Метод оперує

множиною моделей атак соціадьної інженерії, які використовуються як база знань про поведінку атак в розподіленій комп'ютерній системі.

Класифікатор атак працює на двох рівнях: на першому рівні він визначає, чи має вхідна інформація ознаки соціальної інженерії; на другому рівні уточнюється оцінка для визначення типу (класу) атаки та конкретної тактики, що застосовується ініціатором атаки.

Важливу роль відіграє компонент моніторингу стану (Condition Monitoring Component, CMC)), який накопичує всі зібрані показники та забезпечує цілісне представлення поточного контексту взаємодії. Це запобігає надлишковим перевіркам, зберігає історію зібраних ознак та динамічно оновлює ймовірнісну оцінку належності повідомлення до заданої категорії атаки.

Доповнюючи це, компонент механізму моделювання (Simulation Engine Component, SEC)) моделює реакції користувачів або поведінку ініціатора атаки, створюючи умови для навчання агентів у середовищі, що характеризується високою невизначеністю та різноманітністю можливих стратегій ініціатора атаки.

Вся система задумана як ієрархічна структура, в якій агент високого рівня вирішує, які конкретні дії призначити спеціалізованим субагентам нижчого рівня. Ці дії можуть включати аналіз структури URL-адрес, перевірку репутації домену, семантичну оцінку повідомлення або виявлення індикаторів психологічного тиску в голосових дзвінках [27 – 29]. Таким чином, формується багаторівнева система прийняття рішень, яка може гнучко реагувати на нові тактики соціальної інженерії та знижувати ризики для розподіленої комп'ютерної системи шляхом проактивного збору відповідних ознак та поступового зменшення ентропії інформаційного стану.

Метод включає використання компонента менеджера взаємодій (Interaction Manager) (IMC) – повноцінного агента, здатного до цілеспрямованої взаємодії для ведення діалогів з інформаційним середовищем.

Агент τ отримує від компонента моніторингу стану поточну конфігурацію середовища, яка відображає набір індикаторів та поведінкових характеристик, зібраних на даний момент. Цей стан потім відображається на вибір дії α_τ з набору можливих дій A відповідно до політики $\rho(\alpha \mid \sigma(\tau))$.

Компонент механізму моделювання (Simulation Engine Component) генерує відповідь, яка служить основою для обчислення сигналу винагороди ω_τ, яка повертається агенту. В результаті середовище переходить у новий стан $\sigma(\tau + 1)$, і таким чином розгортає послідовність ітерацій, що імітують реальну динаміку комунікації.

Мета агента полягає в пошуку оптимальної політики, яка забезпечує максимізацію очікуваної кумулятивної винагороди. Винагорода ω_τ визначається функцією, $\Omega(\sigma(\tau), \alpha_\tau)$ де коефіцієнт $\lambda \in [0,1]$ служить дисконтним коефіцієнтом, що зменшує вагу віддалених у часі результатів.

Стан середовища описує множину зібраних індикаторів атаки, до яких можуть належати ознаки фішингових повідомлень, лінгвістичні маркери психологічного тиску, метадані електронної пошти або характерні патерни URL-адрес.

Позначимо A як множину типів атак соціальної інженерії, а I – множину індикаторів, пов'язаних з цими атаками. Тоді стан $\sigma(\tau)$ можна подати як вектор $[\chi_1, \chi_2, \dots, \chi_{|I|}, \tau, \varepsilon]$, де кожен компонент χ_i є закодованим тернарним представленням наявності індикатора: підтверджена наявність, підтверджена відсутність або невизначений стан.

Вектор включає параметр τ, що відображає поточний крок діалогу, та змінну ε , яка сигналізує про повторення попередніх запитів. Ця формалізація дає змогу методу не лише накопичувати знання про зібрані індикатори, але й адаптивно керувати стратегією запитів, мінімізуючи інформаційну надмірність і підвищуючи точність виявлення атак соціальної інженерії в умовах неповної та суперечливої інформації.

Простір дій агента формується як об'єднання двох підмножин, що представляють різні типи дій у процесі виявлення атак соціальної інженерії.

Визначаємо $A = T \cup I$, де T – множина типів атак, I – множина індикаторів, які можна запитувати для уточнення контексту повідомлення або взаємодії. Якщо вибрана дія α_τ належить до I, агент ініціює запит додаткової інформації щодо конкретного індикатора, наприклад, перевірку на наявність підозрілих доменів, характерних шаблонів I URL $A = T \cup I$ – адрес, лексичних маркерів психологічного тиску або сигналів аномальної поведінки користувача. У разі вибору дії з підмножини T агент формує остаточне рішення щодо ймовірного типу атаки, що завершує процес взаємодії в рамках поточного сценарію [30 – 31].

Успішність процесу оцінюється за точністю ідентифікації загрози, тобто тим, наскільки правильно агент класифікував конкретну взаємодію до певного класу соціальної інженерії. Така формалізація дає змогу агенту динамічно балансувати між активним збиранням ознак та ухваленням рішень, оптимізуючи взаємодію з інформаційним середовищем для досягнення максимальної ефективності в умовах неповної або суперечливої інформації.

Водночас ієрархічна структура системи забезпечує гнучкість у розподілі обчислювальних ресурсів і сприяє поступовому зменшенню невизначеності під час оцінювання ризику кожного повідомлення чи взаємодії, що є критично важливим для стійкості розподіленої комп'ютерної системи до атак соціальної інженерії.

Синтез архітектури комп'ютерної системи, стійкої до атак соціальної інженерії, на основі ієрархічної багатоагентної системи

Позначимо множину атак соціальної інженерії як множину, T поділену на K групи, де $T = T_1 \vee T_2 \vee \ldots \vee T_K$, підмножини не перетинаються, тобто $T_i \wedge T_j = \emptyset$ для $i \neq j, i, j = 1, \ldots, K$. Кожна підмножина T_{i} відповідає певній категорії атак або тактичній стратегії порушника та пов'язана з власною множиною індикаторів I_{i}, які характеризують ознаки повідомлень, поведінкові патерни чи технічні сигнали, притаманні цій групі. Така організація дає змогу агентам нижчого рівня зосереджуватися на конкретних типах загроз, не витрачаючи ресурси на аналіз менш релевантних сигналів, водночас зберігаючи модульність і масштабованість архітектури.

Ієрархічна структура мультиагентної системи передбачає два рівні: – агент з прийняття рішень вищого рівня (Decision Agent, агент прийняття / ухвалення рішень, центральний агент, DA), який координує загальну політику збору інформації та прийняття рішень; – агенти обслуговування нижчого рівня або сервісні агенти (Service Agents, SA) відповідають кожен за власну підмножину атак T_{i} та пов'язані з нею індикатори I_{i} .

Сервісні агенти виконують функції локального аналізу, ініціюють запити до користувачів або системних сенсорів, оцінюють атрибути повідомлень і передають результати координатору верхнього рівня. Завдяки такому розподілу функцій агент ухвалення рішень може формувати обґрунтовані рішення на основі агрегованих сигналів, спрямованих на мінімізацію ризику та зменшення невизначеності щодо характеру потенційної атаки.

Вибір дій агентами формалізується через відображення стану середовища $\sigma(\tau)$ на простір дій α_τ з множини $A = T \cup I$, де кожна дія нижчого рівня стосується лише підмножини індикаторів I_{i}, а дії агента верхнього рівня координують послідовність запитів і формування підсумкової оцінки типу загрози. Така ієрархічна організація дає змогу системі поступово накопичувати релевантні ознаки, адаптивно коригувати стратегію збирання даних і забезпечувати ефективне виявлення складних сценаріїв соціальної інженерії навіть в умовах неповної або суперечливої інформації.

У кожному циклі взаємодії центральний агент отримує стан середовища, $\sigma(\tau)$ який відображає сукупність усіх зібраних індикаторів потенційних атак соціальної інженерії на цей момент. На основі політики ρ_0 $(\alpha_0\tau/\sigma(\tau)$ агент вищого рівня приймає рішення щодо подальшого курсу дій: продовжувати збір ознак через запити до користувача чи системних датчиків, або формувати остаточну оцінку ймовірного типу загрози. Простір дій агента прийняття рішень визначається множиною $A_0 = \{W_1, W_2, \ldots, W_K, C\}$, де W_{i} позначає сервісного агента, відповідального за аналіз певної підмножини індикаторів I_{i}, а C відповідає модулю класифікації атак.

Коли активовано одного з агентів обслуговування (Service Agents), ініціюється підзадача, в рамках якої $\tau \in [\tau_0, \tau_0 + L]$ той самий агент обслуговування виконує наступні кроки збору ознак, де L визначає максимальну кількість ітерацій для підзадачі. Поточна підзадача завершується після отримання конкретної відповіді на запит щодо індикатора, яка може набувати дискретних значень, наприклад «підтверджено» або «не підтверджено». Якщо активується модуль класифікації C, агент верхнього рівня формує підсумкове рішення щодо типу атаки, завершуючи поточний цикл взаємодії [27].

Агент ухвалення рішень координує роботу всіх сервісних агентів, оптимізує послідовність запитів та процес прийняття рішень, тоді як сервісні агенти зосереджуються на локальних підзадачах зі збирання ознак, що суттєво знижує обчислювальну складність і підвищує точність виявлення складних сценаріїв соціальної інженерії. Водночас така архітектура системи дає змогу агентам адаптивно змінювати свою стратегію залежно від поточного стану середовища, мінімізуючи невизначеність і підвищуючи стійкість розподіленої IT – архітектури до атак соціальної інженерії.

Коли сервісного агента активовано, він отримує стан середовища $\sigma(\tau)$ та витягує відповідні індикатори υ_τ, що відповідають призначеній йому підгрупі загроз $\upsilon_\tau = [\chi_1, \chi_2, \ldots, \chi_{|I_I|}, \tau, \varepsilon]$.

Кожен сервісний агент працює в межах локального підпростору ознак, що дає йому змогу зосереджуватися на окремих категоріях атак соціальної інженерії та відповідних поведінкових або технічних сигналах. На основі локальної політики, сформованої компонентом Interaction Manager (IMC) $\rho_i(\alpha_i\tau/\upsilon_\tau)$, сервісний агент обирає наступний індикатор для уточнення або перевірки та виконує дії зі збирання додаткової інформації, яка уточнює ймовірність конкретного типу атаки.

Простір дій кожного сервісного агента визначається множиною $A_i = \upsilon_1, \upsilon_2, \ldots, \upsilon_{|I_i|}$, де кожен елемент відповідає певному індикатору або шаблону, який агент може запитувати або перевіряти в рамках своєї підзадачі. Така локалізація дій дає змогу сервісним агентам поступово накопичувати знання щодо конкретної категорії загроз, зменшуючи вплив від менш релевантних сигналів і забезпечуючи більш точну оцінку ризику [28].

Ієрархічна взаємодія з агентом арийняття рішень дозволяє агрегувати ці локальні результати та коригувати загальну політику системи, підвищуючи адаптивність та стійкість розподіленої IT – архітектури до атак соціальної інженерії навіть за складних умов неповної чи суперечливої інформації.

В ієрархічній архітектурі з підкріплювальним навчанням, агент прийняття рішень отримує зовнішню винагороду $\omega_e\tau$ на кожному циклі взаємодії, яка відображає ефективність прийнятих рішень у виявленні потенційних атак соціальної інженерії. Після виконання підзавдання, агент прийняття рішень розраховує накопичену винагороду $\Omega_0\tau$ за даний цикл, враховуючи внесок кожного сервісного агента та результати локальної оцінки показників [29 – 31].

Нехай K позначає кількість сервісних агентів, а $\lambda \in [0,1]$ – коефіцієнт дисконтування, який зменшує вагу віддалених у часі результатів. Тоді накопичена винагорода визначається як

$$: \Omega_0\tau = \begin{cases} \sum_{\kappa=1}^{L} \lambda^{\kappa}\omega^{e}_{(\tau+\kappa)}, if\, \alpha_0\,\tau = W_{\text{i}}; \\ \omega^{e}_{(\tau)}, if\, \alpha_0\tau = C. \end{cases}, \quad (1)$$

де W_{i} – сервісний агент, відповідальний за підзадачу аналізу підмножини індикаторів I_{i}, C – модуль класифікації типів атак.

Така формалізація дає змогу агенту прийняття рішень інтегрувати результати локального аналізу, оцінювати ефективність дій на всіх рівнях та адаптивно коригувати політику збору даних і прийняття рішень. Використання коефіцієнта дисконту λ забезпечує баланс між короткостроковими та довгостроковими наслідками, що є критично важливим для підвищення точності виявлення складних сценаріїв соціальної інженерії.

Мета функції $L(\Theta_0)$ визначається через дискретну динаміку навчання, що ґрунтується на підкріплювальному сигналі та історії взаємодій. Нехай σ' –наступний стан середовища після виконання дії $\alpha_0{}'$, $\Theta_0{}'$ – параметри цільової мережі, Θ_0 – параметри поточної мережі та B_0 – буфер досвіду агента прийняття рішень.

Тоді функція втрат визначається як:

$$L(\Theta_0) = E_{\sigma,\alpha_0,\Omega_0,\sigma'\sim B_0}((\Omega_0 + \lambda max_{\alpha_0'} Q_0(\sigma', \alpha_0'; \Theta_0') - Q_0(\sigma', \alpha_0; \Theta_0))^2), \quad (2)$$

де $Q_0(\sigma, \alpha_0; \Theta_0)$ оцінює очікувану кумулятивну винагороду згідно з поточною політикою та $\lambda max_{\alpha_0'} Q^0(\sigma', \alpha^{0\prime}; \Theta^{0\prime})$ забезпечує врахування довгострокових наслідків дій. Така формалізація дає змогу агенту прийняття рішень ефективно інтегрувати інформацію від сервісних агентів та модуля класифікації, оцінюючи як короткострокові, так і довгострокові наслідки своїх рішень, що є критично важливим для підвищення точності виявлення складних атак соціальної інженерії. Сервісні агенти отримують внутрішню винагороду $\omega_i\tau$ на кожному циклі взаємодії, яка оцінює ефективність їх локальної роботи з підмножиною індикаторів I_{i}, закріпленою за певною категорією атак соціальної інженерії.

Метою сервісних агентів є максимізація цих винагород, що забезпечує точне та ефективне виконання підзадач зі збору інформації. Функція втрат кожного сервісного агента $L(\Theta_{\text{i}})$ визначається на основі локальної політики та історії взаємодії в буфері відтворення B_i, де $\Theta_i{}'$ – параметри цільової мережі, Θ_{i} – параметри поточної мережі:

$$L(\Theta_i) = E_{\sigma_i,\alpha_i,\Omega_i,\sigma'\sim B_i}((\Omega_i + \lambda max_{\alpha_i'} Q_i(\sigma_i', \alpha_i'; \Theta_i') - Q_0(\sigma_i, \alpha_i; \Theta_i))^2), \quad (3)$$

де $Q_{\text{i}}(\upsilon_{\text{i}}, \alpha_{\text{i}}; \Theta_{\text{i}})$ – очікувана кумулятивна винагорода за поточною локальною політикою, $\lambda max_{\alpha_i'} Q_i(\upsilon_i', \alpha_i'; \Theta_i')$ – довгострокові наслідки дій агентів.

Така формалізація дає змогу сервісним агентам адаптивно коригувати власну стратегію збору ознак, зосереджуючись на локальних підзадачах, і поступово підвищувати точність оцінювання ризиків для конкретних категорій атак. Координація з агентом прийняття рішень дає можливість агрегувати локальні результати та забезпечує оптимізацію загальної політики системи.

Механізм розрахунку ентропійної винагороди в ієрархічній багатоагентній системі

В рамках ієрархічної архітектури системи, починаючи з початкового стану σ_0, для основного класифікатора загроз та допоміжних класифікаторів, що спеціалізуються на окремих категоріях атак соціальної інженерії, формуються початкові ймовірнісні розподіли. Для агента прийняття рішень позначимо цей розподіл як $\phi_0^D = [\phi_1, \phi_2, \dots, \phi_k]$, де кожен компонент ϕ_i відповідає апостеріорній оцінці належності поточного сценарію до певного класу загроз. Для сервісних агентів визначимо аналогічний розподіл $\phi_0^S = [\phi_1, \phi_2, \dots, \phi_j]$, де кожен елемент представляє ймовірність виявлення відповідної підкатегорії атаки в межах множини можливих дій A.

Важливою характеристикою цих розподілів є ентропія, яка відображає ступінь невизначеності під час прийняття рішень. Початкова ентропія агента прийняття рішень, що працює на рівні інтегральної оцінки загрози, визначається виразом : $J^D(\sigma_0) = -\sum_{\iota=1}^{\kappa} - \phi_\iota \log(\phi_\iota)$, тоді як початкова ентропія сервісних агентів розраховується за формулою : $J^S(\sigma_0) = -\sum_{\iota=1}^{|A|} \phi_\iota \log(\phi_\iota)$.

Обидві метрики ентропії [32 – 36] виконують роль нормалізаторів, запобігаючи зміщенням у значеннях винагороди під час різних епізодів взаємодії. Це особливо важливо у ситуаціях, коли характер інформації, що надходить від користувачів або внутрішніх компонентів системи, може суттєво відрізнятися між сценаріями. Таким чином, використання ентропії як керувального параметра надає ієрархічній

багатоагентній системі можливість адаптивно балансувати між точністю виявлення атак та стійкістю до інформаційного шуму, що є характерним для умов роботи реальних розподілених ІТ – архітектур.

Аналогічно, у момент часу τ обчислюється інформаційна ентропія основного класифікатора агента прийняття рішення та $J^D(\sigma_\tau)$ класифікатора $J^S(\sigma_\tau)$ агента обслуговування . Після того, як система виконує дію та переходить у новий стан $\sigma_{\tau+1}$, оновлені значення ентропії визначаються як $J^D(\sigma_{\tau+1})$ та $J^S(\sigma_{\tau+1})$.

Різниця між цими значеннями стає основою для розрахунку винагороди, яка характеризує зменшення невизначеності в роботі агента прийняття рішення та агентів обслуговування. Для агента прийняття рішення значення винагороди визначається співвідношенням:

$$\rho^D_{\text{entropy}} = max(\frac{(J^D(\sigma_\tau) - J^D(\sigma_{\tau+1}))}{J^H}(\sigma_0), 0). \quad (4)$$

Розмір винагороди сервісних агентів визначається за формулою:

$$\rho^S_{\text{entropy}} = max(\frac{(J^S(\sigma_\tau) - J^S(\sigma_{\tau+1}))}{J^S}(\sigma_0), 0). \quad (5)$$

У цих виразах відношення різниці ентропій до початкового значення виконує нормалізацію функціонала, даючи системі змогу коректно порівнювати рівень інформаційного приросту між різними епізодами взаємодії. Використання оператора max запобігає появі від'ємних значень, оскільки на практиці можливі ситуації, коли через неповні дані або зашумлені сигнали ентропія збільшується, а отже, невизначеність не зменшується. Такий крок забезпечує стабільне навчання, спрямоване на поступове зниження інформаційного хаосу, що є характерним для атак соціальної інженерії, коли порушники свідомо створюють ситуації когнітивної плутанини.

За такої структури винагороди ієрархічна багатоагентна система отримує додатковий механізм самоадаптації, спрямований на пошук рішень, що мінімізують ентропію стану, поступово переводячи КС з невизначеного стану до такого, що має чіткий діагностичний висновок щодо наявності або відсутності індикаторів атаки.

Зовнішню функцію [36 – 40] винагороди ρ_t^{ext} визначимо наступним чином:

$$\rho_t^{ext} = \sum_{\iota=1}^{n} N + \rho^D_{\text{entropy}}, \quad (6)$$

де N є додатним числом, якщо агент видає остаточне рішення, і це рішення є правильним (умова успіху виконана); N є від'ємним числом, якщо вибрана дія на кроці t, повторює попередньо виконану дію в тому ж епізоді (виявлено повторення дії, надлишкові запити караються), або якщо досягнуто максимально дозволеної кількості кроків взаємодії без остаточного рішення (бюджет епізоду вичерпано, нерішучість карається); N дорівнює 0 у всіх інших випадках (взаємодія триває, винагорода пропорційна зменшенню невизначеності, додаткова вигода від інформативних дій, що зменшують невизначеність, масштабується) $\rho^D_{\text{entropy}} > 0$.

Система внутрішніх винагород у ієрархічній багатоагентній моделі

Внутрішня винагорода ρ_t^{int} для сервісного агента визначається наступним чином:

$$\rho_t^{\text{int}} = \sum_{\iota=1}^{n} N + \rho^S_{\text{entropy}}, \quad (7)$$

де N є додатним числом, якщо запит сервісного агента дає остаточне та правильне рішення цільового індикатора або ознаки (спостерігається правильне збіг); N є від'ємним числом, якщо срвісний агент видає повторний або надлишковий запит, який не приносить нової інформації (повторення дій); N дорівнює 0 у всіх інших випадках (поступове зменшення невизначеності без остаточного збігу) $\rho^S_{\text{entropy}} > 0$.

Максимальна кількість кроків взаємодії встановлює верхню межу тривалості епізоду та використовується в умові тайм-ауту для зовнішньої винагороди. Нормалізація за початковою ентропією [41 - 43] для обох рівнів забезпечує порівнюваність винагород у різних сценаріях з різним рівнем початкової невизначеності. Компонент «success» узгоджує поведінку агента верхнього рівня з вимогою точної фінальної класифікації, тоді як локальний компонент «match» спрямовує сервісних агентів на точне та ефективне уточнення ознак. Штрафи за повторювані дії на обох рівнях виконують роль регуляризаторів, які запобігають циклам і надмірним запитам.

У сукупності такі визначення збалансовують прагнення до короткої та рішучої взаємодії з необхідністю формального й кількісно вимірюваного зменшення діагностичної невизначеності.

Ієрархічна структура винагород забезпечує узгоджений розподіл функцій між рівнями системи: агент ухвалення рішень виступає координатором оцінювання безпеки розподіленої ІТ – архітектури, а підпорядковані агенти виконують функції локальних фільтрів. Це забезпечує формування багаторівневого захисту від атак соціальної інженерії, знижує ризики когнітивної маніпуляції та підтримує ефективну адаптацію системи в динамічних умовах інформаційного середовища.

Окрім винагороди, що базується на зменшенні невизначеності ентропії, основний агент отримує додаткові підкріплення, які генеруються залежно від результату прийнятого рішення. Якщо остаточна класифікація сценарію атаки успішна, тобто система правильно ідентифікує спробу соціальної інженерії або підтверджує її відсутність, то застосовується позитивний приріст винагороди, який позначається як σ^+. У разі некоректної ідентифікації або досягнення максимально допустимої кількості ітерацій діалогу вводиться штрафна складова σ^-, яка відображає неефективність стратегії агента [44 – 46].

Для сервісних агентів, що виконують локалізовані завдання аналізу індивідуальних поведінкових особливостей користувачів, винагорода визначається за іншим принципом . Якщо обраний агент вірно уточнює ознаку, що характеризує можливу атаку, та отримує однозначну відповідь від середовища, то він отримує позитивний приріст τ^+. Якщо відбувається повторний запит, який не зменшує ентропію інформаційного стану та призводить до інформаційного шуму, то вводиться негативний штраф τ^-.

Формалізація сигналів підкріплення на різних рівнях архітектури системи поєднує глобальну оцінку продуктивності головного агента з локальною точністю роботи підлеглих агентів. Використання параметрів σ^+, σ^-, τ^+ та τ^- дозволяє збалансувати стратегічні та тактичні аспекти функціонування системи, створюючи умови для адаптивного навчання в динамічному середовищі.

У контексті протидії атакам соціальної інженерії це означає, що головний агент навчається розрізняти загальну структуру маніпулятивних сценаріїв, тоді як сервісні агенти зосереджуються на дрібніших деталях взаємодії, які разом формують цілісну та стійку поведінкову модель ІТ – архітектури.

Формування підграфів знань та їх інтеграція в процес прийняття рішень агентами

У межах запропонованої архітектури системи знання про сценарії атак соціальної інженерії подаються у формі графа знань, інтегрованого в компонент Interaction Manager. У цьому графі вузли відповідають індикаторам атак та їх проявам, а також класам атакувальних дій, тоді як ребра описують кореляційні залежності між цими елементами. Кожному ребру призначається вага, яка характеризує ступінь взаємозв'язку між конкретним індикатором та класом атаки; значення ваги визначається на основі ймовірнісної статистики, отриманої під час аналізу навчальних і тестових сценаріїв.

Оскільки метод використовує ієрархічну багатоагентну структуру, сервісні агенти відповідають за обробку підмножин можливих індикаторів та класів атак. Для кожного такого агента будується окремий підграф знань, що містить набір індикаторів O_i та набір атак A_i, що стосуються його підсистеми. Це забезпечує локалізацію обчислень та зменшує загальну складність синтезу архітектури, оскільки окремі агенти спеціалізуються на певних класах загроз.

Вагові коефіцієнти ребер у підграфах трактуються як умовні ймовірності появи певної ознаки за умови реалізації відповідного класу атаки. Для формалізації такої залежності вводиться наступне співвідношення:

$$q(o_u a_v) = \frac{m(a_v, o_u)}{\sum_{i=1}^{n} m(a_v, o_z)}, \tag{8}$$

де $m(a_v, o_u)$ – кількість зареєстрованих сценаріїв, у яких атака типу a_v супроводжується проявом ознаки o_u. Обчислюючи значення ймовірності для кожної пари «атака–ознака», можна побудувати матрицю умовних ймовірностей $\Lambda_i \in R^{|O_i| \times |A_i|}$, яка використовується в процесі навчання агента.

Така формалізація дає змогу відобразити складну багатовимірну структуру залежностей між різними типами атак соціальної інженерії та їх ознаками, що забезпечує можливість адаптивного навчання агентів у процесі взаємодії з потенційно ворожими сценаріями.

На основі матриці умовних ймовірностей, введеної раніше, поточний розподіл ймовірностей проявів ознак для сервісних агентів визначається як:

$$\pi(o_u) = \Lambda_i \cdot \pi(a_v), \tag{9}$$

де $\pi(a_v) \in R^{|A_i| \times 1}$ – вектор ймовірностей відповідних класів атак, який обчислюється підпорядкованим класифікатором для підгрупи i, та $\pi(o_u) \in R^{|O_i| \times 1}$– розподіл ймовірностей для локальних ознак, що стосуються цього агента. Ця операція перетворює ймовірності класів атак на ймовірності прояву конкретних ознак, забезпечуючи основу для прийняття рішень агентом.

Далі цей розподіл ймовірностей поєднується з оцінками Q значень, обчислених сервісними агентами, що дозволяє враховувати як попередні знання про ймовірності появи ознак, так і досвід навчання агента в динамічному середовищі. Формально вибір наступної дії агента виражається як : $\alpha_\tau = argmax_\omega(\sigma(Q_\omega(\sigma_\tau)) + \sigma(\pi(o_u)))$, де $Q_\omega(\sigma_\tau)$ розподіл Q– значень, обчислених сервісним агентом для стану σ_τ а σ – сигмоїдна функція нормалізації, яка відображає значення в діапазон [0,1], функція argmax визначає клас ознак або дій, на якому агент зупиняє свій вибір для подальшого запиту або оцінки.

Завдяки такому поєднанню ймовірностей та Q значень , агент отримує можливість враховувати як попередній досвід, так і оновлену інформацію про поведінку користувача або системи, що критично важливо для протидії атакам соціальної інженерії. Такий метод дозволяє агенту адаптивно вибирати пріоритетні

ознаки для перевірки, підвищуючи точність виявлення та зменшуючи ймовірність помилкових оцінок у багаторівневій системі.

Дослідження ефективності методу синтезу розподіленої комп'ютерної системи, стійкої до атак соціальної інженерії

Для оцінки ефективності запропонованої ієрархічної мультиагентної архітектури було досліджено чотири аспекти: точність виявлення та класифікації порівняно з одноагентними та неієрархічними підходами; ефективність, тобто вплив адаптивних ентропійних запитів на тривалість взаємодії та затримку; здатність до узагальнення між різними каналами соціальної інженерії (електронна пошта, фішингові вебсторінки, імперсонація в OSN, голосовий вішинг); абляції для оцінки внеску ієрархії, графа знань та ентропійної складової винагороди.

Експерименти виконувалися у середовищі Python 3.13 із використанням PyTorch 3.x, оптимізованого для сучасних GPU. Обчислення здійснювалися на NVIDIA RTX 5090, за підтримки CUDA 13 та cuDNN 10. Токенізація тексту й рендеринг вебсторінок здійснювалися через beautifulsoup4, lxml та Playwright 1.50 у headless-режимі з локальними снапшотами, що усувало мережеву варіативність.

Агент прийняття рішень реалізовано як Deep Q-Network із окремою цільовою мережею, коефіцієнтом дисконту 0,99, буфером досвіду 200 000 переходів, мінібатчем 256 і оптимізатором Adam зі швидкістю 3×10^{-4}. Пошук здійснювався за ε-евристикою: ε зменшувався від 0,20 до 0,01 протягом 200 000 кроків.

Сервісні агенти реалізовано як спеціалізовані модулі для електронної пошти та вебмодальності: текстові дані оброблялися BiLSTM, вебознаки комбінацією CNN та MLP, що отримують DOM-структури, URL та метадані. Максимальний бюджет взаємодії становив п'ять дій на приклад.

Попередні знання інтегрувалися шляхом ініціалізації логів сервісних агентів умовними ймовірностями $P(feature \mid attack)$, отриманими з графа знань. Дії, що суперечили цим залежностям, маскувалися. Функцію винагороди доповнювали ентропійним терміном, що стимулював зниження невизначеності, та штрафами за повторення або малозначущі дії, сприяючи формуванню коротких інформативних послідовностей.

Запропоновану архітектуру порівнювали з: системою на основі правил; найкращими одномодальними моделями (URL, текст, DOM); плоскою DQN-моделлю без ієрархії; ансамблем із різнім злиттям; а також абляційними варіантами без ієрархічної структури, без пріорних графових знань або без ентропійної складової винагороди.

У методі використано мультимодальний корпус, що містить електронні листи та веб-сторінки, анотовані відповідно до типової таксономії атак соціальної інженерії: фішинг, вішінг, клонування профілю, грумінг, spear phishing, троянські програми, baiting, scareware, hoaxing, water-holing, приманювання програмним забезпеченням, маскування файлів, а також атаки, представлені у текстовому вигляді (tailgating, dumpster-diving тощо). Для категорій, що не мають природних сенсорних вимірювань, ознаки було подано як текстові описи в листах або на веб-сторінках.

Корпус містить **80 000 зразків**: – 48 000 електронних листів (24 тис. шкідливих, 24 тис. легітимних), – 32 000 веб-сторінок (16 тис. шкідливих, 16 тис. легітимних).

Дані розподілено на вибірки **70% / 10% / 20%** у хронологічному порядку для запобігання інформаційному витоку. Забезпечено роздільність доменів веб-ресурсів та відправників листів. Дублі або майже дублікати вилучені за допомогою SimHash для тексту, перцептивного гешування для зображень та нормалізації URL-адрес. Виконано автоматичне визначення мови, маскування персональних даних та розв'язання редіректів для отримання кінцевої сторінки. Для веб-сторінок додатково виділялися структурні токени DOM-дерева та дії над формами.

Такі кроки забезпечують уніфікацію даних, зменшення хибних кореляцій та узгоджений процес вилучення ознак у поштовій і веб-модальностях.

Ієрархічна архітектура з верхньорівневим агентом, що вибирає відповідного сервісного агента для аналізу модальності, демонструє вищу точність, кращі робочі характеристики та меншу кількість дій на приклад порівняно з неієрархічними базовими моделями за однакового бюджету взаємодії.

Експериментальна оцінка ефективності ієрархічної мультиагентної архітектури

Ієрархічна архітектура з верхньорівневим агентом прийняття рішень, який координує вибір модальності відповідного сервісного агента, демонструє вищу точність, кращі робочі характеристики та меншу кількість дій на приклад порівняно з неієрархічними базовими моделями за однакового бюджету взаємодії.

У таблиці 1 наведено площу під кривою характеристик приймача (AUC-ROC), площу під кривою precision–recall (AUC-PR), F1-міру, частку хибнопозитивних спрацювань при 95% істиннопозитивній частці, кількість дій на зразок та медіанну наскрізну затримку (end-to-end latency) у мілісекундах.

Таблиця 1

Бінарне виявлення

Метод	**Площа під ROC–кривою**	**Площа під кривою PR**	**F1–оцінка**	**Рівень хибнопозитивних результатів**	**Кількість кроків взаємодії**	**Затримка (мс)**
Система на основі правил	0,78	0,62	0,65	0,42	–	–
Найкраща одномодальна модель	0,90	0,85	0,84	0,18	–	–
Плоска глибока Q–мережа (один агент)	0,92	0,88	0,86	0,15	6,1	210
пізнє ансамблювання	0.93	0.90	0.88	0.13	–	–
Запропонований метод	0.96±0.01	0.94 ± 0.01	0.92 ± 0.01	0.08 ± 0.01	3.2 ± 0.2	155

У порівнянні з базовою моделлю у вигляді одноагентної Deep Q-Network, ієрархічний метод підвищує площу під кривою характеристик приймача на +0.04, зменшує частку хибнопозитивних спрацювань при 95% істиннопозитивній частці на −0.07 та скорочує кількість дій на зразок приблизно на 47% (з 6.1 до 3.2), що демонструє одночасне покращення як точності, так і ефективності.

Було проведено оцінювання багатокласової продуктивності системи відповідно до таксономії типів атак соціальної інженерії.

Оскільки деякі категорії трапляються відносно рідко, для коректного узагальнення результатів було використано макроусереднене значення метрики F1 (macro–averaged F1). Такий метод забезпечує рівну вагу кожного класу, незалежно від частоти його появи в наборі даних.

Результати експериментів наведено окремо для двох модальностей: електронної пошти та вебсередовища, що дозволяє порівняти ефективність класифікації у різних контекстах виявлення атак соціальної інженерії. Узагальнені показники подано в Таблиці 2.

Таблиця 2.

Результати для модальностей електронної пошти та вебсередовища

Модальність	**Найкраща неієрархічна базова лінія**	**Запропонована ієрархічна структура (агент прийняття рішень + агенти обслуговування)**	**Різниця**
Email	0,77	0,86 ± 0,01	+0,09
Web	0,74	0,84 ± 0,02	+0,10

Ієрархічна політика, що керується головним агентом прийняття рішень і підтримується службовими агентами, ініціалізованими графом знань, забезпечує приріст від +9 до +10 пунктів за макроусередненою метрикою F1 (macro–F1) порівняно з найкращими неієрархічними альтернативами в обох модальностях (електронна пошта та веб). Це свідчить про більш надійне розпізнавання рідкісних тактик соціальної інженерії.

Для того щоб ізолювати внесок саме ієрархічного управління, було проведено аналіз абляції, у межах якого всі інші компоненти – енкодери, параметри оптимізації та архітектурні налаштування – залишалися незмінними.

Таким чином, здійснювалося порівняння повної системи з її одноагентною конфігурацією, що дозволило кількісно оцінити ефект ієрархічного керування. Таблиця 3 узагальнює вплив вилучення ієрархії на показники точності та ефективності системи.

Таблиця 3.

Результати абляційного аналізу впливу ієрархії

Варіант	Площа під кривою ROC (AUC ROC)	Макроусереднене значення F1 (Macro–F1)	Кількість дій на зразок	Рівень хибних спрацювань при 95% TPR
Повна система (агент прийняття рішень + службові агенти + апріорні знання на основі графа знань + механізм винагород із урахуванням ентропії).	0,96	0,92	3,2	0,08
Без ієрархії (одноагентна конфігурація)	0,93	0,88	5,7	0,12

Ієрархічна структура, що поєднує агента прийняття рішень та службових агентів, є основним джерелом підвищення ефективності системи, забезпечуючи зменшення кількості дій на один зразок у 2,5 раза.

Водночас спостерігається покращення показників продуктивності на суворих порогах роботи: рівень хибних спрацювань при 95% істинно позитивних спрацьовувань знижується з 0,12 до 0,08.

У межах обох завдань бінарного виявлення атак та детальної класифікації тактик і типів дій атак соціальної інженерії на основі об'єднаного корпусу електронних листів і вебвзаємодій, ієрархічна архітектура, у якій агент прийняття рішень координує роботу модально – специфічних службових агентів, стабільно перевершує одноагентні та інші неієрархічні базові моделі.

Запропонований метод досягає вищої точності при меншій кількості кроків взаємодії, забезпечуючи кращі робочі характеристики на практично значущих рівнях істинно позитивних спрацьовувань, що підтверджує ефективність ієрархічного механізму координації агентів у сиснтезу розподіленої КС в контексті виявлення атак соціальної інженерії.

ВИСНОВКИ З ДАНОГО ДОСЛІДЖЕННЯ І ПЕРСПЕКТИВИ ПОДАЛЬШИХ РОЗВІДОК У ДАНОМУ НАПРЯМІ

У проведеному дослідженні обґрунтовано необхідність переходу від традиційних централізованих або одноагентних систем виявлення атак соціальної інженерії до архітектур, здатних працювати в умовах високої гетерогенності, багатомодальності та динамічності сучасних розподілених комп'ютерних систем (КС). Аналіз існуючих підходів показав, що мультимодальні методи, багатоагентні системи та механізми навчання з підкріпленням забезпечують помітне підвищення точності, зменшення хибнопозитивних рішень та покращення адаптивності, проте залишаються обмеженими в аспектах узгодження дій, інтерпретованості та масштабованості.

Запропонований метод синтезу архітектури розподіленої ІТ-системи, стійкої до атак соціальної інженерії, спрямований на подолання цих обмежень шляхом поєднання ієрархічної багатоагентної моделі, графа знань та оптимізаційних принципів навчання з підкріпленням. Така інтеграція дозволяє формувати узгоджену глобальну політику на основі локальних рішень вузлів, забезпечувати зменшення ентропії інформаційного стану, скорочувати кількість дій, необхідних для прийняття обґрунтованого рішення, та підвищувати стійкість системи до складних соціально-інженерних сценаріїв.

Результати аналізу доводять, що запропонований метод створює підґрунтя для інженерії масштабованих, адаптивних і високоефективних засобів захисту, які можуть бути інтегровані в сучасні комп'ютерні системи та використовуватися для підвищення рівня безпеки в реальних експлуатаційних умовах.

References

[1] N. Gaddam, "AI-Driven Social Engineering Attack Detection in Email Communications," International Journal of Artificial Intelligence and Deep Learning Research and Development, vol. 6, no. 1, pp. 15–23, 2025.

[2] P. H. Kyaw, J. Gutierrez, and A. Ghobakhlou, "A Systematic Review of Deep Learning Techniques for Phishing Email Detection," Electronics, vol. 13, no. 19, p. 3823, 2024.

[3] J. L. Wilk-Jakubowski, L. Pawlik, G. Wilk-Jakubowski, and A. Sikora, "Machine Learning and Neural Networks for Phishing Detection: A Systematic Review (2017–2024)," Electronics, vol. 14, no. 18, p. 3744, 2025.

[4] M. Murhej and G. Nallasivan, "Multimodal Framework for Phishing Attack Detection and Mitigation," Frontiers in Communications and Networks, 2025.

[5] A. Vulfin, A. Sulavko, V. Vasiliev, A. Minko, A. Kirillova, and A. Samotuga, "A Multimodal Phishing Website Detection System Using Explainable AI Technologies," Preprints.org, 2025, doi: 10.20944/preprints202511.1683.v1.

[6] Y. Li, H. Hu, Z. Zhang, and Y. Chen, "KnowPhish: LLM-based Multimodal Reference-Based Phishing Detection," in USENIX Security Symposium, 2024.

[7] C. Liu, S. Wang, Z. Chen, L. Huang, Y. Li, and H. Li, "KGhish: A Phishing Website Detection Method Based on Knowledge Graph," in ICIC 2024, Springer, pp. 300–311, 2024.

[8] S. Asiri, Y. Xiao, S. Alzahrani, and M. Ju, "PhishingRTDS: A Real-time Detection System for Phishing Attacks Using a Deep Learning Model," Computers & Security, 2024, 103843.

[9] A. Alzu'bi, B. B. Gupta, and A. Almomani, "A Multimodal Deep Learning Framework for Phishing Detection Using Textual and Visual Features," Computers & Security, vol. 141, p. 103518, 2024.

[10] L. Wang, T. Zhang, and Y. Chen, "Email Threat Detection via Transformer-Based Multimodal Fusion," IEEE Access, vol. 12, pp. 115560–115574, 2024.

[11] H. Wang, X. Du, and S. Xu, "Deep Semantic–Visual Fusion for Phishing Webpage Detection," Information Fusion, vol. 104, 102317, 2024.

[12] J. Zhang and K. Liu, "Cross-Modal Phishing Detection Using Graph Contrastive Learning," Pattern Recognition, vol. 156, 110856, 2025.

[13] Y. Xue, E. Spero, Y. S. Koh, and G. Russello, "MultiPhishGuard: An LLM-based Multi-Agent System for Phishing Email Detection," arXiv:2505.23803, 2025.

[14] W. Li, S. Manickam, Y.-W. Chong, and S. Karuppayah, "PhishDebate: An LLM-Based Multi-Agent Framework for Phishing Website Detection," arXiv:2506.15656, 2025.

[15] T. T. Cao, C. Huang, Y. Li, H. Wang, A. He, N. Oo, and B. Hooi, "PhishAgent: A Robust Multimodal Agent for Phishing Webpage Detection," AAAI, vol. 39, no. 27, pp. 35003–35011, 2025.

[16] I. Mohiuddin and A. Almogren, "Ensemble Techniques for Detecting Profile Cloning Attacks in Online Social Networks (OSNs)," PeerJ Computer Science, e3182, 2025.

[17] M. Soltani, K. Khajavi, M. Jafari Siavoshani, and A. H. Jahangir, "A Multi-Agent Adaptive Deep Learning Framework for Online Network Intrusion Detection," Cybersecurity, vol. 3, no. 1, pp. 199–210, 2024.

[18] B. Buyuktanir, Ş. Altinkaya, G. Karatas Baydogmus, and K. Yildiz, "Federated Learning in Intrusion Detection: Advancements, Applications, and Future Directions," Springer, 2025.

[19] Y. Javed, M. A. Khayat, A. A. Elghariani, and A. Ghafoor, “PRISM: A Hierarchical Intrusion Detection Architecture for Large-Scale Cyber Networks,” IEEE Transactions on Dependable and Secure Computing, vol. 20, no. 2, pp. 345–358, 2023.

[20] F. Alhayan, A. Alshuhail, A. O. A. Ismail, O. Alrusaini, S. Alahmari, A. E. Yahya, S. S. Albouq, and M. Al Sadig, “Enhanced Anomaly Network Intrusion Detection Using an Improved Snow Ablation Optimizer with Dimensionality Reduction and Hybrid Deep Learning Model (EAID-OADRHM),” Scientific Reports, vol. 15, no. 1, 1199655, 2025.

[21] A. Gudnavar, K. Naregal, and B. Madagouda, “Cyber Threat Detection and Analysis Using Dual-Layered Hierarchical Capacity Particle Filtering (HCPF),” Journal of Computer Information Systems, 2025.

[22] L. Diana, P. Dini, and D. Paolini, “Overview on Intrusion Detection Systems for Computers and Networks,” Applied Sciences, vol. 15, no. 1, p. 123, 2025.

[23] M. Rafferty and J. Higgins, “LLM-Assisted Incident Response for Social Engineering Attacks,” Computers & Security, vol. 142, 103545, 2024.

[24] Q. Yan, X. Wang, L. Chen, and F. R. Yu, “Graph-Based Anomaly Detection for Enterprise-Scale IT Systems,” IEEE Transactions on Network and Service Management, vol. 21, no. 1, pp. 111–124, 2024.

[25] S. Sharma and P. Pokharel, “ML-Driven Email Threat Hunting in Distributed Systems,” Future Generation Computer Systems, vol. 157, pp. 450–465, 2024.

[26] S. Ren, H. Guo, Z. Li, X. Wang, and J. Zhang, “ARCS: Adaptive Reinforcement Learning Framework for Automated Cybersecurity Incident Response,” Applied Sciences, vol. 15, no. 2, p. 951, 2025.

[27] A. V. Singh, E. Rathbun, E. Graham, L. Oakley, S. Boboila, A. Oprea, and P. Chin, “Hierarchical Multi-Agent Reinforcement Learning for Cyber Network Defense,” arXiv:2410.17351, 2024.

[28] T. Hürten, T. Gülmez, A. Stüber, P. Babarczi, and D. Böttger, “Hierarchical Multi-Agent Reinforcement Learning for Autonomous Cyber Defense in Coalition Networks,” in IEEE MILCOM, 2024.

[29] Z. He, Y. Liu, J. Wang, and W. Chen, “Multi-Agent Risk-Sensitive Reinforcement Learning with Reward Shaping for Cyber Defense,” IEEE Transactions on Information Forensics and Security, vol. 19, pp. 1452–1465, 2024.

[30] D. Tang, Q. Wang, and S. Zhu, “Hierarchical Multi-Agent Reinforcement Learning with Intrinsic Penalties for Repeated Actions,” in IJCAI, 2024.

[31] K. Prakasam and S. Kannan, “A Hierarchical RL Model for Threat Hunting in Large-Scale IT Systems,” Computers & Security, vol. 143, 103632, 2025.

[32] M. Espinoza and A. Böttger, “Cooperative Multi-Agent Reinforcement Learning for Enterprise Network Defense,” IEEE Access, vol. 13, pp. 137234–137249, 2024.

[33] S. Ghosh and Y. Wang, “Hierarchical Policy Learning for Distributed Security Monitoring,” Information Sciences, vol. 660, 121973, 2025.

[34] P. Bera, “Multi-Agent RL for Behavioral Threat Intelligence Aggregation,” Expert Systems with Applications, vol. 245, 123051, 2025.

[35] A. Kim and R. Sutton, “Adaptive Hierarchical Agents for Large-Scale Decision-Making,” Journal of Machine Learning Research, vol. 26, pp. 1–35, 2025.

[36] Z. Fei, Y. Xu, Z. Han, and L. Qian, “Entropy-Regularized Reinforcement Learning for Decision-Making Under Uncertainty,” IEEE Transactions on Neural Networks and Learning Systems, vol. 35, no. 4, pp. 4567–4581, 2024.

[37] A. Shibata, T. Matsushima, and T. Fujita, “Variational Policy Optimization with Adaptive Entropy Regularization,” AAAI Conference on Artificial Intelligence, 2024.

[38] H. Zhang, Z. Yang, and M. J. Kochenderfer, “Uncertainty-Aware Reinforcement Learning via Information Gain Rewards,” Machine Learning, vol. 113, pp. 1987–2011, 2024.

[39] K. Lee, A. Raghunathan, and P. Liang, “Stochastic Value Gradients with Entropy-Based Regularization,” arXiv:2403.11249, 2024.

[40] C. Xu, W. Wang, T. Lan, and S. Subramaniam, “Entropy-Driven Multi-Agent Reinforcement Learning for Cyber Defense Automation,” Computers & Security, vol. 140, 103072, 2025.

[41] A. Jaques, J. Garcia, N. Brown, and B. Eysenbach, “Intrinsic Motivation in Reinforcement Learning: A Survey,” Foundations and Trends in Machine Learning, vol. 17, no. 1, pp. 1–150, 2024.

[42] A. R. Mahmood, M. Agarwal, and R. Sutton, “Hierarchical Reinforcement Learning with Intrinsic Rewards for Skill Acquisition,” Journal of Machine Learning Research, vol. 25, pp. 1–34, 2024.

[43] A. Singh and B. Vasan, “Entropy-Based Reward Shaping for Sequential Threat Detection,” Knowledge-Based Systems, vol. 296, 110987, 2024.

[44] T. Wu and R. Chandra, “Hierarchical Information-Gain Rewards for Multi-Agent Exploration,” Neural Networks, vol. 174, pp. 106–118, 2025.

[45] Y. Inoue and F. Zhong, “Uncertainty-Aware Policy Learning for High-Stakes Decision Environments,” Pattern Recognition, vol. 158, 111012, 2025.

[46] W. Sun, A. K. Singh, and J. Peters, “Reward Decomposition and Intrinsic Exploration for Large-Scale Multi-Agent Systems,” Sensors, vol. 25, no. 3, p. 812, 2025.