Вінницький національний технічний університет

Факультет менеджменту та інформаційної безпеки

Кафедра менеджменту та безпеки інформаційних систем

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

Підвищення захищеності системи розумний дім засобами штучного інтелекту на основі мережі LSTM

Виконав: ст. 2-го курсу, групи 2КІТС-22м
спеціальності 125– Кібербезпека
Освітня програма – Кібербезпека
інформаційних технологій та систем

(шифр і назва напряму підготовки, спеціальності)

Дроганов Д. О.
(прізвище та ініціали)

Керівник: к.т.н., доц., доцент каф. МБІС
Сачанюк-Кавецька Н. В
(прізвище та ініціали)
«04» грудня 2023 р.

Опонент: к.т.н., доц., доцент каф. ОТ
Черняк. О.І.
(прізвище та ініціали)
«04» грудня 2023 р.

Допущено до захисту
Голова секції УБ кафедри МБІС

Юрій ЯРЕМЧУК
“04” грудня 2023 р.

Вінниця ВНТУ - 2023 рік

Вінницький національний технічний університет
Факультет менеджменту та інформаційної безпеки
Кафедра менеджменту та безпеки інформаційних систем

Рівень вищої освіти ІІ-й (магістерський)
Галузь знань 12 – Інформаційні технології
Спеціальність 125 – Кібербезпека
Освітньо-професійна програма - Кібербезпека інформаційних технологій та систем

ЗАТВЕРДЖУЮ
Голова секції УБ, кафедра МБІС

_____ **Юрій ЯРЕМЧУК**
“20” вересня 2023 р.

З А В Д А Н Н Я
на магістерську кваліфікаційну роботу студенту
Дроганову Дмитру Олександровичу
(прізвище, ім`я, по-батькові)

1. Тема роботи Підвищення захищеності системи розумний дім засобами штучного інтелекту на основі мережі LSTM

Керівник роботи к.т.н., доцент Сачанюк-Кавецька Наталія Василівна
(прізвище, ім`я, по-батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу від “18” вересня 2023 року № 247
2. Строк подання студентом роботи
3. Вихідні дані до роботи: технічне завдання, схеми, методика розрахунків

4. Зміст текстової частини: Анотація; зміст; вступ; основні теоретичні принципи та аналіз системи «розумний дім»; проектування інтелектуальної системи безпеки на основі LSTM; експериментальне дослідження та оцінка ефективності алгоритму ідентифікації та нейтралізації кіберзагроз у системі «розумний дім»; економічна частина; исновки; список використаних джерел, додатки

5. Перелік ілюстративного матеріалу (з точним зазначенням обов`язкових креслень)
Презентація Power Point (слайд)

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Основна частина	Сачанюк-Кавецька Н. В., к.т.н., доцент, доцент каф. ВМ		
Економічна частина	Причепа І.В., к.е.н., доц. каф. ЕПВМ		

7.Дата видачі завдання 20 вересня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи		Примітка
1	Визначення напрямку магістерської роботи, формулювання теми	20.09.2023	25.09.2023	
2	Аналіз предметної області обраної теми	26.09.2023	28.09.2023	
3	Апробація отриманих результатів	29.09.2023	02.10.2023	
4	Розробка алгоритму роботи	03.10.2023	17.10.2023	
5	Написання магістерської роботи на основі розробленої теми	18.10.2023	10.11.2023	
6	Розробка економічної частини	11.11.2023	23.11.2023	
7	Передзахист магістерської кваліфікаційної роботи	24.11.2023	25.11.2023	
8	Виправлення, уточнення, корегування магістерської кваліфікаційної роботи	26.11.2023	30.11.2023	
9	Захист магістерської кваліфікаційної роботи	15.12.2023	15.12.2023	

Студент _____ Дрогалів Д.Д.
(підпис)

Керівник роботи _____ Сачанюк-Кавецька
(підпис)

АНОТАЦІЯ

УДК 621.391

Дроганов Д. О. Підвищення захищеності системи розумний дім засобами штучного інтелекту на основі мережі LSTM. Магістерська кваліфікаційна робота зі спеціальності 125 – «Кібербезпека», освітня програма «Кібербезпека інформаційних технологій та систем». Вінниця: ВНТУ, 2023. 132 с.

На укр. мові. Бібліогр.: 42 назви; рис.: 30; табл. 5.

У цій магістерській кваліфікаційній роботі розглянуто питання підвищення захищеності систем «Розумний дім» за допомогою штучного інтелекту на основі мереж LSTM. В роботі детально досліджено існуючі архітектури розумних домів, основні компоненти інтелектуальних систем, а також сучасні методи забезпечення безпеки. Проаналізовано принципи роботи та класифікацію мереж LSTM, оцінено потенційні загрози та ризики у системах розумних домів. Основна увага приділена розробці математичної моделі та алгоритмів для ідентифікації та класифікації потенційних загроз. Проведено експериментальне дослідження та оцінка ефективності запропонованого рішення. Робота також містить аналіз економічної ефективності науково-технічної розробки.

Ключові слова: розумний дім, штучний інтелект, мережі LSTM, кібербезпека, ідентифікація загроз.

ABSTRACT

UDC 621.391

Drohanov D. O. Increasing the security of the smart home system by means of artificial intelligence based on the LSTM network. Master's qualification thesis on specialty 125 - «Cybersecurity», educational program «Cybersecurity of information technologies and systems». Vinnytsia: VNTU, 2023. 132 p.

In Ukrainian speech Bibliography: 42 titles; Fig.: 30; table 5.

In this master's qualification work, the issue of increasing the security of «Smart Home» systems with the help of artificial intelligence based on LSTM networks is considered. The work examines in detail the existing architectures of smart homes, the main components of intelligent systems, as well as modern methods of ensuring security. The working principles and classification of LSTM networks were analyzed, potential threats and risks in smart home systems were assessed. The main attention is paid to the development of a mathematical model and algorithms for the identification and classification of potential threats. An experimental study and evaluation of the effectiveness of the proposed solution was carried out. The work also contains an analysis of the economic efficiency of scientific and technical development.

Keywords: smart home, artificial intelligence, LSTM networks, cyber security, threat identification.

ЗМІСТ

ВСТУП

У сучасному світі, де технології розвиваються стрімко, системи «Розумний дім» набувають все більшої популярності, забезпечуючи користувачам зручність та автоматизацію побутових процесів. Однак, цей поступ супроводжується зростаючими загрозами у сфері кібербезпеки. Зі збільшенням кількості підключених пристроїв та інтеграцією їх у єдину мережу, вразливості цих систем стають більш привабливими цілями для кіберзлочинців. Це ставить на порядок денний важливе питання про розробку ефективних методів захисту «Розумний дім» від потенційних кібератак.

Штучний інтелект, зокрема мережі LSTM, являють собою прогресивні інструменти, що здатні аналізувати та прогнозувати поведінку користувачів, вчасно ідентифікувати незвичайні або підозрілі патерни, які можуть свідчити про кібернапади. Ці мережі здатні вивчати та запам'ятовувати довготривалі залежності, що дозволяє їм виявляти загрози, базуючись не лише на одноразових подіях, але й на послідовності дій протягом тривалого часу. Це робить LSTM ідеальним вибором для розумних систем, де потрібно швидко реагувати на динамічно мінливі умови та забезпечувати безперервний захист [1].

Дослідження, що базується на глибокому аналізі існуючих викликів у сфері безпеки «розумних» систем, пропонує інноваційний підхід до розв'язання цих проблем за допомогою адаптивних алгоритмів. Використання LSTM у системах «розумний дім» може забезпечити не лише попередження кібератак у реальному часі, але й адаптацію до нових загроз, які ще не були раніше ідентифіковані. Такий підхід відкриває шлях до створення більш надійних та інтелектуальних систем безпеки, що спроможні захистити користувачів від непередбачених кіберзлочинів і становлять собою значний крок вперед у сфері кібербезпеки розумного житла.

Актуальність теми полягає у зростаючій потребі забезпечення високого рівня конфіденційності та цілісності даних в системах «розумний дім», які стають все більш поширеними в Україні. З огляду на стрімкий розвиток Інтернету речей та збільшення числа підключених пристроїв, існує величезний потенціал для зловмисних втручань, що робить захист цих систем критично важливим.

За допомогою критичного аналізу існуючих рішень та порівняння їх з потенційними можливостями мереж LSTM, дослідження підтверджує важливість розробки нових методів захисту. Це стане значним внеском у сферу кібербезпеки та зміцнить позиції України в області застосування передових технологій у повсякденному житті громадян.

Актуальність даної теми безсумнівна та зумовлена кількома ключовими чинниками, які мають вирішальне значення для розвитку галузі кібербезпеки в Україні та у всьому світі. Зростання числа «розумних» пристроїв у приватних домогосподарствах та на підприємствах веде до збільшення кількості потенційних векторів для кібератак. Такі системи часто включають в себе компоненти, вразливі до зовнішніх загроз, що можуть бути використані для несанкціонованого доступу або навіть для шкідливих дій проти інфраструктури.

Традиційні методи захисту, такі як антивірусне програмне забезпечення та файрволи, вже не можуть забезпечити адекватний захист від сучасних загроз, які постійно еволюціонують та стають більш складними. Це призводить до потреби у впровадженні більш просунутих технологій, здатних адаптуватися та протистояти новим викликам.

Також, розвиток мереж LSTM у контексті кібербезпеки «розумного дому» є перспективним напрямком, який може забезпечити більшу захищеність системи за рахунок їх здатності до глибокого навчання та аналізу поведінкових патернів. Це створює основу для розробки інтелектуальних систем безпеки, які можуть передбачати та нейтралізувати потенційні загрози навіть перед тим, як вони матимуть місце.

Для України, яка переживає швидкий технологічний розвиток та одночасно зіштовхується з різноманітними кіберзагрозами, впровадження таких передових систем є особливо актуальним. Розробка вітчизняних рішень у сфері кібербезпеки не лише зміцнить національну безпеку, але й сприятиме розвитку вітчизняного високотехнологічного сектору, створенню нових робочих місць, та підвищенню інноваційного іміджу України на міжнародній арені.

Робота корелює з державними програмами України, які спрямовані на розвиток інформаційних технологій та кібербезпеки, враховуючи національні інтереси в сферах оборони, економіки та інфраструктури. Зокрема, вона відповідає основним напрямкам «Стратегії кібербезпеки України», спрямованої на створення надійних та безпечних інформаційних систем, які є ключовими для забезпечення стійкості критичної інфраструктури країни.

Крім того, дослідження вписується в плани наукових установ та вищих навчальних закладів, які займаються розробкою новітніх технологій у сфері ШІ, штучного інтелекту та машинного навчання. Впровадження результатів роботи може зміцнити дослідницьку базу, сприяти залученню інвестицій у наукові дослідження та розвиток інноваційних проектів.

З огляду на галузеві та державні програми, робота може сприяти реалізації планів з підвищення ефективності енергоспоживання, телекомунікацій та інших ключових секторів економіки, які все більше інтегруються з концепцією «розумного дому».

Метою дослідження є створення і валідація інноваційної інтелектуальної системи безпеки для розумних будинків, здатної ефективно протистояти сучасним кіберзагрозам за допомогою технологій штучного інтелекту на основі мереж LSTM.

Основні задачі, які необхідно вирішити для досягнення цієї мети, включають:

– аналіз теоретичної бази для інтелектуальних систем безпеки «розумних будинків», включаючи аналіз існуючих концепцій та методів, оцінку ризиків та недоліків сучасних систем безпеки;

– проектування концептуальних засад і методик інтелектуальної системи безпеки, визначення її фундаментальних функцій та елементів, а також розроблення механізму роботи алгоритму ідентифікації потенційних загроз;

– реалізація проектованої системи, включаючи імплементацію ключових модулів, детальну специфікацію розроблених алгоритмів, конфігурацію та налаштування системи;

– перевірка та валідація ефективності проектованої інтелектуальної системи безпеки через експериментальне тестування та аналіз ризиків, а також оцінку її практичної придатності та визначення шляхів подальшого вдосконалення.

Об'єктом дослідження є процеси та явища, пов'язані з кіберзахистом систем «розумний дім».

Предметом дослідження є розробка та валідація інноваційної інтелектуальної системи безпеки для «розумних будинків», здатної протистояти сучасним кіберзагрозам.

Наукова новизна одержаних результатів полягає у розробленні нової концепції інтелектуальної системи безпеки для розумних будинків на базі LSTM мереж. Зокрема, вперше одержано комплексну модель, яка дозволяє системі «розумний дім» аналізувати поведінкові патерни користувачів і адаптуватися до них для прогнозування та попередження потенційних кіберзагроз.

Практична новизна виражається у створенні алгоритму для ідентифікації та нейтралізації кіберзагроз в реальному часі, що було удосконалено завдяки використанню LSTM мереж. Система не лише відповідає на вже відомі загрози, але й здатна прогнозувати нові, невідомі атаки, вчасно адаптуючи захисні механізми.

Наукове використання результатів можливе в рамках подальших досліджень у сфері кібербезпеки, інформаційних технологій та штучного інтелекту, а також може бути рекомендовано для академічних курсів, що вивчають розвиток і застосування інтелектуальних систем.

З практичної точки зору, результати дослідження готові до імплементації в системи «розумний дім» на ринку України, що може сприяти підвищенню рівня безпеки житла та, в ширшому сенсі, - рівня кібербезпеки в країні.

Економічний ефект від впровадження цих розробок може бути виражений у зниженні витрат на відновлення систем після кібератак, зменшенні ризику витоку конфіденційної інформації та підвищенні довіри користувачів до технологій «розумного дому».

1 ОСНОВНІ ТЕОРЕТИЧНІ ПРИНЦИПИ ТА АНАЛІЗ СИСТЕМИ «РОЗУМНИЙ ДІМ»

1.1 Історія та концепція розумних будинків

Концепція «Розумного Дому» є втіленням прогресивної ідеї інтелектуальної автоматизації житлового простору. Ця концепція базується на використанні передових досягнень в області інженерії та інформаційних технологій (рис. 1,1). Система «Розумного Дому» інтегрує в себе різноманіття пристроїв – від простих сенсорів температури до складних мультимедійних систем. Така інтеграція направлена на створення синергії між технічними елементами, що, в свою чергу, забезпечує підвищення рівня комфорту, безпеки та енергоефективності житла.

Рисунок 1.1 – Концепція «Розумного Дому»

В основі «Розумного Дому» лежить принцип максимальної автономії: система здатна самостійно управляти енергопостачанням, клімат-контролем, освітленням, забезпеченням безпеки, а також багатьма іншими аспектами домашнього господарства. Це досягається шляхом впровадження комплексних алгоритмів, які дозволяють системі аналізувати поточні умови та історію взаємодії з мешканцями, прогнозувати їх потреби та відповідно адаптувати свою роботу.

«Розумний Дім» не тільки спрощує побутові процеси, але й вносить вклад у підвищення якості життя мешканців. Забезпечуючи ефективне управління ресурсами, така система допомагає знизити енергоспоживання та оптимізувати витрати. Ця інноваційна система є результатом еволюції традиційного житла у відповідь на виклики сучасності, що ставить на перший план екологічність,

економічність та індивідуальний підхід до потреб кожного мешканця.. Її особливості охоплюють:

Система захисту в контексті інтелектуального житла відіграє роль гаранта безпеки, як фізичного, так і матеріального благополуччя його мешканців. Вона охоплює наступні елементи:

– сенсорна мережа. Фундаментальну основу безпекової системи становлять датчики, що перехоплюють варіації умов проживання – від руху та відкриття вікон до нехарактерних температурних змін чи наявності небезпечних газів;

– обробка інформації. Процесори обробляють інформацію, отриману від сенсорів, аналізуючи її та видаючи команди для активізації захисних механізмів.

– відеоаналітика. Системи відеонагляду не тільки фіксують події, а й аналізують поведінку осіб, ідентифікуючи потенційні загрози;

– сигналізація та зв'язок. Аудіо- та візуальні сповіщення, а також комунікаційні механізми забезпечують оперативне інформування мешканців та служб надзвичайних ситуацій;

– інтеграційні вузли. Спеціалізовані шлюзи забезпечують взаємозв'язок усіх компонентів безпеки, інтегруючи їх у єдину систему «Розумного Дому».

Залучення технологій штучного інтелекту, зокрема LSTM-мереж, істотно підсилює потенціал такої системи. Нейронні мережі здатні виявляти складні шаблони у великому потоці даних, що дозволяє прогнозувати потенційні інциденти та реагувати на них заздалегідь. Ця технологія може забезпечити вивчення поведінкових патернів мешканців та відповідно адаптувати параметри безпеки, щоб враховувати особливості кожного індивідуума та зміни в його поведінці, що підвищує загальний рівень захищеності системи.

Архітектура «розумного будинку» може бути реалізована через дві ключові структурні концепції: централізовану і децентралізовану моделі. Відповідно до централізованого варіанту, всі основні елементи системи, такі як механізми керування, центральний процесор і виконавчі модулі, об'єднані в одну єдину телекомунікаційну мережу (рис. 1.2). Така організація дозволяє здійснювати

високоорганізоване і синхронізоване управління, а також безперебійну передачу команд по всій системі.

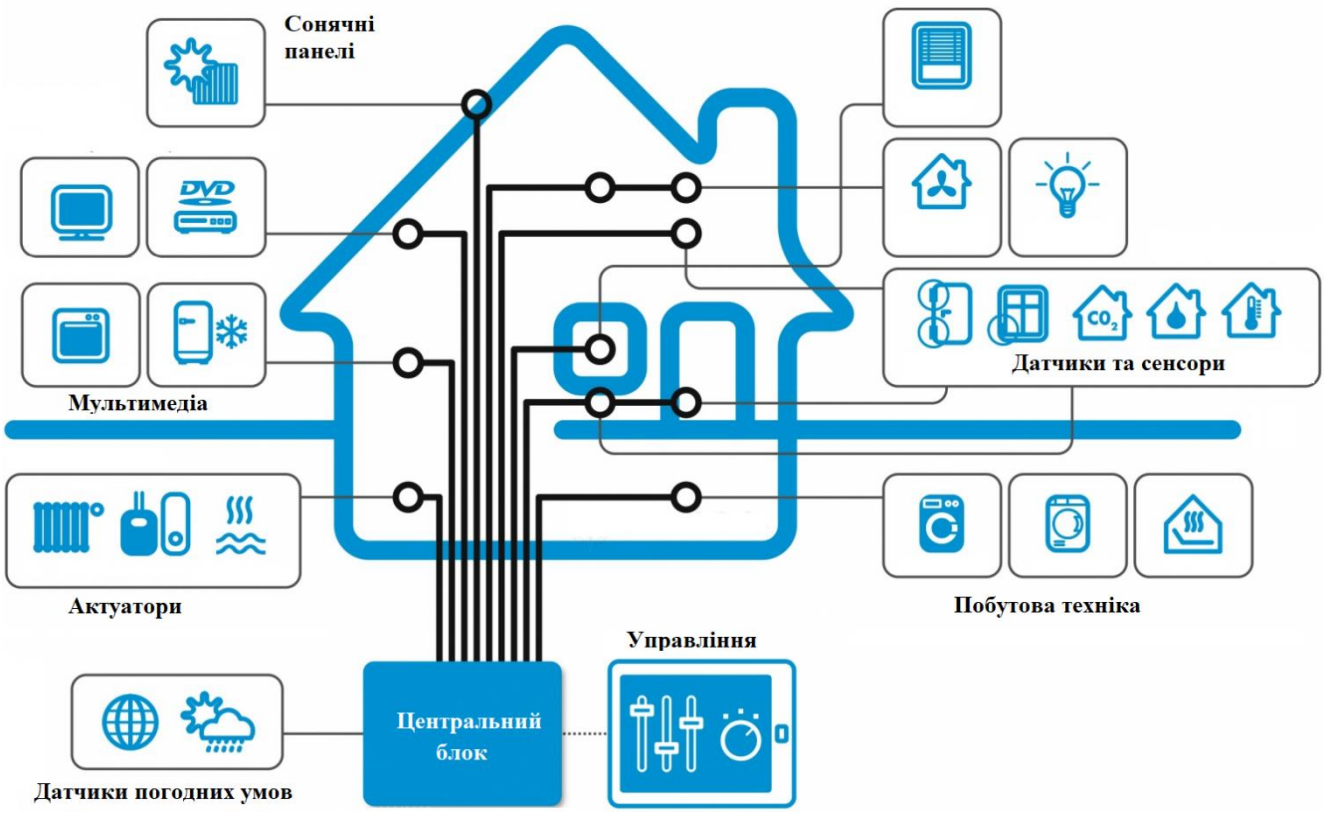


Рисунок 1.2 – Схема централізованої архітектури «Розумний дім»

Інтерфейси управління виступають як місток між користувачем та системою, дозволяючи контролювати функції «розумного будинку» через множину пристроїв – від класичних пультів до сучасних сенсорних панелей і мобільних додатків. Також, автоматизація дозволяє системі самостійно реагувати на зміни оточення за допомогою вбудованих датчиків, що відстежують світло, присутність, температуру, вологість і інші параметри.

Центральний обчислювальний контролер є нервовим центром системи, що відповідає за збір, аналіз, та обробку інформації, яку надають як інтерфейси керування, так і датчики. Він зберігає прописані користувачем сценарії управління, а також може самостійно генерувати дієві команди, базуючись на алгоритмах штучного інтелекту. Контролер координує роботу всіх підсистем «розумного будинку», відправляючи оперативні команди до виконавчих пристроїв для реалізації заданих завдань [3].

Централізована архітектура «розумного будинку» володіє низкою переваг і недоліків, які слід розглянути при виборі системи автоматизації. Зокрема, до її переваг можна віднести:

– уніфікація управління. Централізована система дозволяє інтегрувати всі підсистеми і пристрої в єдиний інтерфейс, що спрощує управління;

– координація дій. Єдиний контрольний центр може ефективно координувати роботу всієї системи, забезпечуючи синхронізовані дії;

– легкість масштабування. Додавання нових пристроїв або модернізація існуючих компонентів може бути здійснена через один центральний вузол;

– зручність моніторингу. Можливість відстежувати стан всіх компонентів системи з одного місця полегшує моніторинг.

– спрощення технічної підтримки: Централізація управління полегшує пошук несправностей та надання технічної підтримки.

Недоліки централізованої архітектури:

– точка єдиного збою. Усі системи залежать від центрального контролера, тому його збій може призвести до відмови всієї системи;

– складність обслуговування. Якщо центральний контролер виходить з ладу, відновлення системи може бути складним і вимагати спеціалізованого обслуговування;

– масштабованість. Незважаючи на те, що додавання нових компонентів є можливим, існує верхня межа, після якої система може вимагати оновлення або заміни центрального контролера;

– вартість. Централізовані системи часто вимагають більших початкових інвестицій, оскільки потрібно обладнання вищої потужності для центрального контролера;

– гнучкість. Установка та налаштування централізованої системи може бути менш гнучкою в порівнянні з децентралізованими аналогами.

Отже, централізована архітектура пропонує ефективне і зручне управління «розумним будинком», але в той же час вона має певні обмеження, особливо коли мова йде про надійність і вартість [4].

У децентралізованій архітектурі інтелектуальних систем управління житлом відходять від ідеї одноосібного центрального контролю на користь розподіленої схеми, де кожен окремий елемент, будь то пристрій або модуль, оснащений власною виконавчою логікою (рис. 1.3). Це означає, що датчики, виявляючи певні події чи зміни в домашньому середовищі, можуть безпосередньо ініціювати активацію виконавчих механізмів, таких як реле або електромеханічні пристрої.

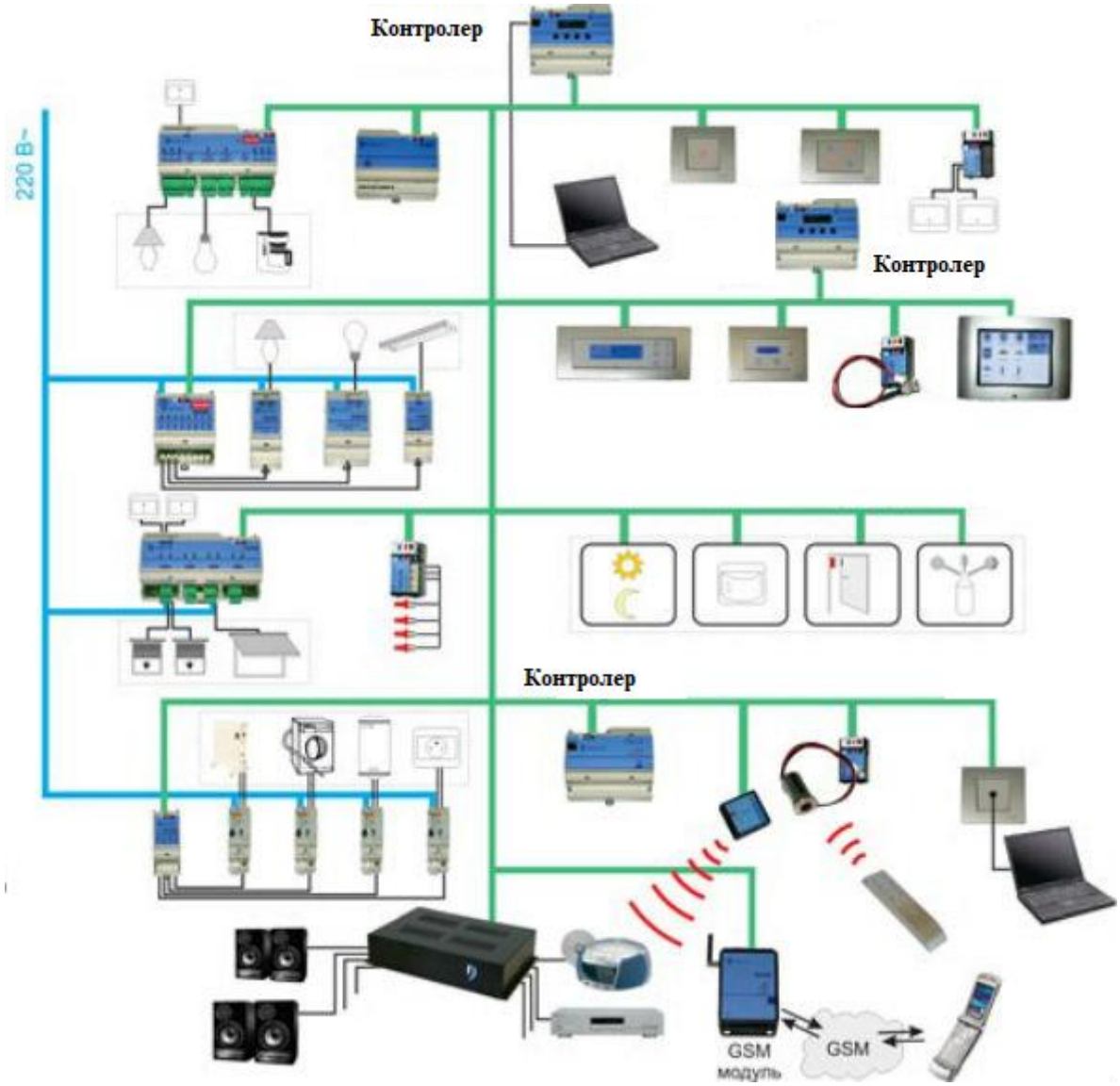


Рисунок 1.3 – Схема децентралізованої архітектури «Розумний дім»

Дана архітектура наділяє систему здатністю до саморегуляції на мікрорівні, що дозволяє кожному компоненту адаптуватись до змін умов навколишнього середовища самостійно. Відповідно, це сприяє підвищенню гнучкості та адаптивності системи, яка може бути налаштована таким чином, щоб відповідати унікальним вимогам та перевагам користувачів.

Однак, необхідність синхронізації та взаємодії між окремими розподіленими компонентами може призвести до складнощів в управлінні та моніторингу загальної системи. Така структура може вимагати більш розвиненої інфраструктури для ефективної взаємодії, а також може створити додаткові виклики для централізованої діагностики та обслуговування системи. У результаті, хоча децентралізована архітектура забезпечує більшу самостійність окремих елементів, вона також може вимагати більш складних рішень для інтеграції та управління цими елементами як єдиним цілим [5].

Децентралізована архітектура «розумного будинку» пропонує інший підхід до управління та інтеграції пристроїв та систем. Як і будь-яка система, вона має свої переваги та недоліки. Серед переваг можна виділити:

– відмовостійкість. У разі збою одного з компонентів, решта системи продовжує функціонувати, що знижує ризик повного виходу системи з ладу;

– гнучкість та масштабованість. Легше додавати, змінювати чи оновлювати окремі компоненти без впливу на всю систему, що забезпечує високий рівень кастомізації;

– розподіл навантаження. Обробка даних відбувається локально на рівні окремих пристроїв, що зменшує навантаження на центральний контролер;

– енергоефективність. Можливість локального керування дозволяє оптимізувати споживання енергії в залежності від потреби в конкретному місці або часі;

– адаптивність. Система може бути більш чутливою до змін умов та швидше реагувати на потреби користувачів.

Недоліки децентралізованої архітектури:

– складність управління. Велика кількість незалежних компонентів може ускладнити централізоване управління та моніторинг;

– ризик несумісності. Різні пристрої та системи можуть мати проблеми з взаємодією, якщо вони не належним чином інтегровані;

– безпека. Розподіленість системи може створити більше потенційних точок для злому чи несанкціонованого доступу;

– складність налаштування. Індивідуальне налаштування кожного компоненту може вимагати більше часу та технічних знань;

– вартість. Висока автономність компонентів може призвести до збільшення вартості системи через необхідність у вбудованих обчислювальних та комунікаційних можливостях для кожного пристрою.

Враховуючи ці аспекти, децентралізована архітектура може бути більш вигідною в умовах, де потрібна висока автономність окремих зон чи пристроїв, а також там, де важлива швидка реакція на зміни умов. Однак, для великих та складних систем, де необхідний централізований моніторинг та управління, варто ретельно зважити потенційні труднощі, які може створити децентралізована архітектура [6].

1.2 Архітектура та компоненти системи «Розумний дім»

Архітектурний дизайн інтелектуальних систем, що ґрунтується на принципах штучного інтелекту, є комплексом, який об'єднує в собі інтерактивні фізичні пристрої IoT, сучасні сенсорні рішення, актуатори та продвинуті мережеві технології (рис. 1.4). Ці компоненти взаємопов'язані з потужними обчислювальними платформами, які використовують алгоритми штучного інтелекту для аналізу зібраних даних і автоматизації управлінських процесів.

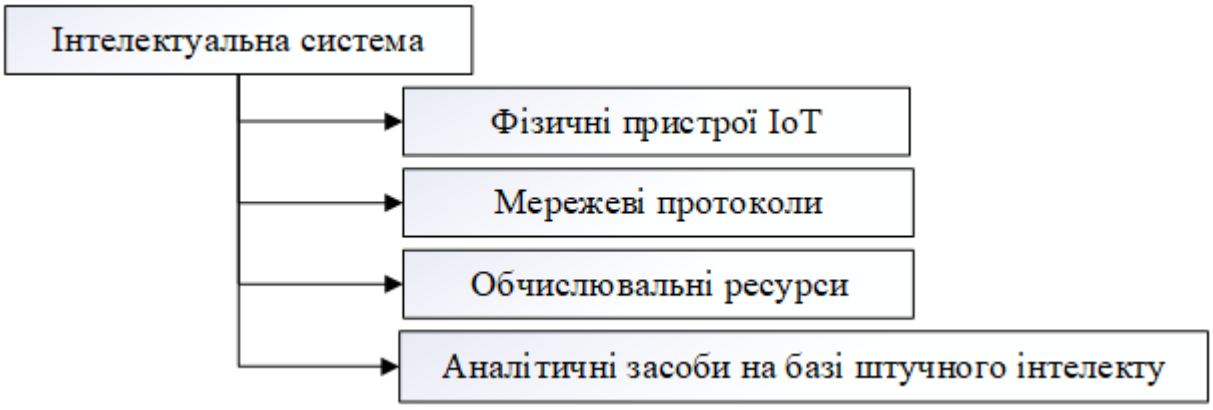


Рисунок 1.4 – Основні компоненти інтелектуальних систем

Як можна побачити із рис. 1.4 основні до основних компонентів архітектури інтелектуальних систем належать:

– фізичні пристрої IoT. Ці пристрої включають широкий діапазон датчиків, що моніторять та реагують на різноманітні аспекти навколишнього середовища, як-от зміни температури, вологості, руху, освітленості, звуку та багато іншого, забезпечуючи системі інформацію про стан фізичного світу;

– мережеві протоколи. Важливою складовою є мережеві протоколи, які можуть використовувати як провідні, так і бездротові технології для забезпечення комунікації між IoT пристроями та центрами обробки даних. Такі протоколи включають Wi-Fi, Bluetooth, LTE та інші технології, які забезпечують передачу даних;

– обчислювальні ресурси. Обчислювальні ресурси можуть бути розташовані локально, на так званих edge серверах, які знаходяться поряд із датчиками та актуаторами, або у віддалених хмарних середовищах, що надають значні обчислювальні потужності для аналізу даних великої складності;

– аналітичні засоби на базі штучного інтелекту. Це включає методи машинного навчання та глибокого навчання, які дозволяють системі ідентифікувати закономірності, аналізувати тренди, прогнозувати майбутні події та автоматично налаштовувати стратегії управління для оптимального реагування на зміни.

Така архітектура сприяє створенню розумного, інтерактивного та високоадаптивного середовища, яке може самостійно реагувати на потреби користувачів і зміни в оточенні, підвищуючи ефективність управління і забезпечуючи вищий рівень автоматизації житлових та комерційних просторів [7].

Сучасна архітектура інтелектуальних систем, що розгортається на фундаменті штучного інтелекту, створює унікальний екосистемний підхід. Вона інтегрує в себе взаємодіючі компоненти, такі як пристрої Інтернету речей, розгалужені мережеві структури та потужні обчислювальні ресурси, орієнтовані на автоматизацію та оптимальне керування процесами в різноманітних оперативних сферах.

На прикладі архітектури RL-IoT, що показана на рис. 1.5 можна спостерігати, як впровадження модулів навчання з підкріпленням спрямоване на досягнення

заздалегідь визначених цілей через допомогу точно налаштованих команд для IoT пристроїв. Ціль тут визначається як кінцевий стан, до якого повинен прийти пристрій або система після проходження ряду етапів і виконання визначених дій.

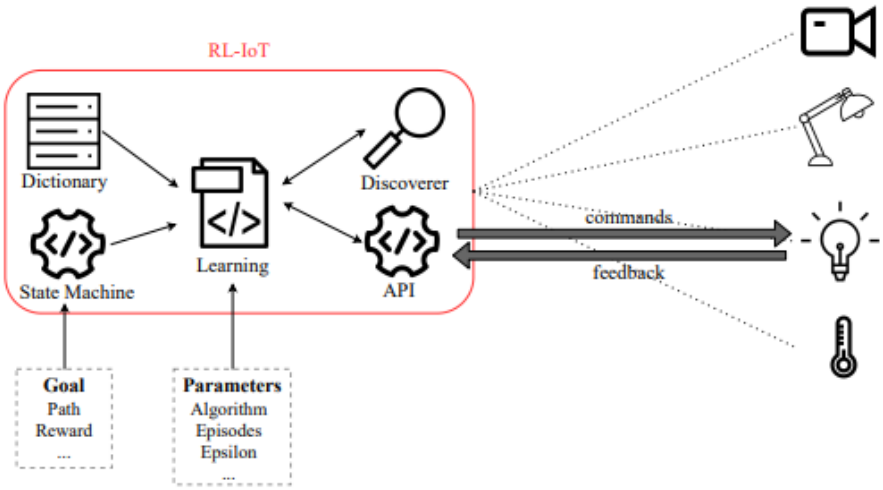


Рисунок 1.5 – Архітектура RL-IoT

У системі RL-IoT, спілкування між пристроями базується на детально розробленому наборі комунікаційних інструкцій, який включає повний набір доступних команд. Цей набір може бути створений за допомогою офіційних протокольних специфікацій, аналізу мережевого трафіку, або методів реверс-інжинірингу.

Завдяки алгоритмам навчання з підкріпленням, як-от Q-Learning чи SARSA, RL модуль системи формує та оновлює внутрішній автомат станів, вибираючи найефективніші дії з комунікаційного лексикону, що спрямовані на досягнення мети. Визначення ефективності дій базується на системі винагороди, яка аналізує потенційний успіх дій на шляху до досягнення поставленої цілі [8-9].

Додаткові модулі, як «Discoverer», відповідають за ідентифікацію IoT пристроїв у локальній мережі, використовуючи методи сканування. Після виявлення пристроїв, модуль Socket API забезпечує необхідну абстракцію комунікаційних механізмів для взаємодії з ними, що дозволяє не лише відправляти команди, але й приймати вхідні дані від пристроїв, відповідно до їх архітектурної здатності приймати такі запити [10].

Інтернет речей (IoT) та автономні системи керування розроблялися як відносно окремі сфери технологій, кожна з власними методами та цілями. IoT зосереджений на підключенні фізичних об'єктів до мережі, тоді як автономні системи фокусуються на самостійному виконанні завдань без людської взаємодії. Об'єднання цих двох напрямків у концепції AIoT (Автономний Інтернет речей) створює новий рівень інтеграції, що може значно розширити можливості та функціональність IoT.

В архітектурі AIoT інтелектуальні пристрої не просто збирають дані через сенсори, але й активно взаємодіють із фізичним світом, використовуючи виконавчі механізми для реалізації рішень, витягнутих із зібраних даних (рис. 1.6). Ця архітектура дозволяє пристроям IoT не тільки підключатися до Інтернету через стаціонарні точки доступу, а й використовувати обмежені обчислювальні можливості мобільних пристроїв або інших точок доступу для первинної обробки даних, перш ніж вони будуть передані на хмарні сервери для подальшого аналізу.

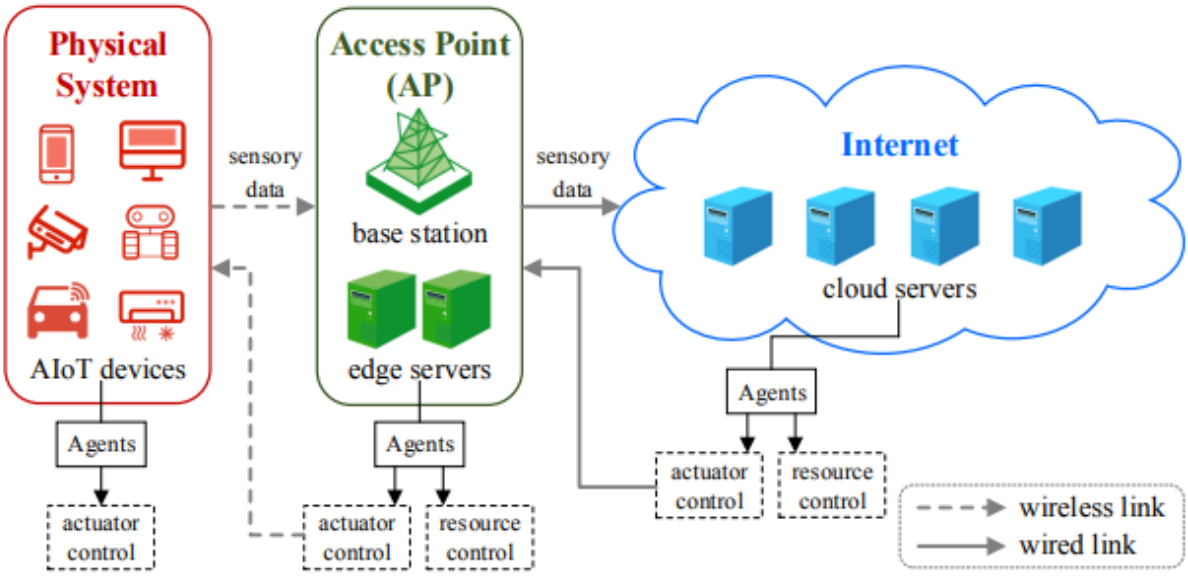


Рисунок 1.6 – Архітектура AIoT

WSAN (Wireless Sensor and Actuator Network), як підсистема AIoT, складається з сенсорів, які збирають інформацію про оточуюче середовище, та виконавчих пристроїв, які взаємодіють із середовищем на основі аналізу цих даних. Все це відбувається через бездротові комунікаційні мережі. З використанням

алгоритмів підсиленого глибокого навчання, автономні агенти у WSAN можуть не тільки обробляти інформацію, але й вирішувати, як керувати виконавчими пристроями, щоб відповідно адаптувати або змінити стан середовища. Це створює динамічну взаємодію, яка дозволяє системі бути відповідною і реактивною до змін у середовищі [11].

В архітектурі AIoT, користувачі стикаються з більш глибоко інтегрованою системою, яка перевищує можливості WSAN, включаючи не тільки збір даних, але й їх аналіз, обробку та виконання. AIoT реалізує комплексний підхід до обробки інформації, що забезпечується завдяки використанню передових комунікаційних технологій та обчислювальних можливостей, що включає в себе широкий спектр сервісів.

Структура AIoT містить три основні шари:

– шар сприйняття. Це основний інтерфейс між фізичним світом та цифровою екосистемою, який включає в себе пристрої Інтернету речей із сенсорами для збору інформації та актуаторами для впливу на оточення;

– мережевий шар. Він відповідає за комунікаційне з'єднання пристроїв IoT між собою та з обчислювальними центрами. Мережевий шар включає в себе різні точки доступу, такі як мобільні базові станції чи Wi-Fi роутери, які служать для передачі даних;

– шар додатків. Цей шар включає в себе сервери та обчислювальні платформи, на яких проводиться аналіз і обробка зібраних даних, а також реалізується автоматизація процесів за допомогою алгоритмів штучного інтелекту.

1.3 Огляд потенційних контролерів для розробки системи «Розумний дім»

Контролери системи «Розумний дім» є центральними елементами автоматизації домашніх систем, які об'єднують і управляють різними розумними приладами в будинку. Вони дозволяють користувачам віддалено керувати освітленням, кліматом, безпекою, та іншими функціями дому через мобільні пристрої, панелі управління, або навіть голосові команди. Також контролери

можуть забезпечувати безпеку дому інтегруючи різноманітні безпекові компоненти, такі як датчики руху, датчики відкриття дверей/вікон, камери спостереження, та системи сповіщення, у єдину мережу. Це дозволяє моніторити та реагувати на будь-які підозрілі дії або надзвичайні ситуації в реальному часі. Користувачі можуть отримувати сповіщення на свої мобільні пристрої при виявленні руху або несанкціонованого входу, що значно підвищує безпеку дому. Найбільш популярними контролерами, що використовуються у системах розумного дому є:

Arduino UNO

Arduino UNO – це одна із найпопулярніших платформ для створення різноманітних електронних проектів, від простих до складних (рис. 1.7). Вона використовується у системах «Розумний дім» для різноманітних застосувань, зокрема, для контролю та автоматизації домашніх процесів.

Рисунок 1.7 – Зовнішній вигляд контролера «Arduino UNO»

Arduino UNO, завдяки своїй гнучкості та легкості програмування, є ідеальним вибором для ентузіастів та професіоналів у галузі «Розумний дім». Ця платформа дозволяє інтегрувати різноманітні модулі та датчики, що робить її універсальним інструментом для розробки індивідуалізованих рішень. Окрім того, Arduino UNO може використовуватися для моніторингу енергоспоживання у будинку, дозволяючи оптимізувати витрати енергії та сприяти сталому споживанню ресурсів. Іншим важливим застосуванням є розробка систем безпеки,

які можуть включати датчики руху, камери спостереження та автоматичні замки дверей, забезпечуючи високий рівень безпеки для домогосподарств.

Таблиця 1.1 – Основні технічні характеристики «Arduino UNO»

Характеристика	Опис
Мікроконтролер	ATmega328P
Робоча напруга	5 В
Вхідна напруга	7-12 В
Цифрові входи/виходи	14 (з яких 6 можуть служити як PWM)
Аналогові входи	6
Постійний струм на вході/виході	20 мА
Постійний струм для 3.3В виходу	50 мА
Частота кристала	16 МГц
USB-з'єднання	Так
ICSP заголовок	Так
Кнопка скидання	Так
Характеристика	Опис

Arduino UNO ідеально підходить для використання в системах «Розумний дім» завдяки своїй універсальності та простоті програмування. Він дозволяє користувачам легко інтегрувати та керувати різними домашніми автоматизаціями, такими як контроль освітлення, температури, безпеки та інші елементи дому. Завдяки своїм чисельним входам та виходам, Arduino UNO є потужним інструментом DIY ентузіастів та майстрів, які шукають доступне та гнучке рішення для створення розумних домашніх систем.

Raspberry Pi Zero

Raspberry Pi Zero є компактною та високоефективною платформою, яка широко використовується для створення інноваційних проектів у сфері електроніки, включаючи системи «Розумний дім» (рис. 1.8). Ця плата є ідеальною для автоматизації домашніх процесів, оскільки дозволяє підключати та інтегрувати різноманітні датчики та пристрої, наприклад, для контролю температури,

вологості, освітлення та безпеки. Можливість підключення до Інтернету через Wi-Fi робить Raspberry Pi Zero особливо зручним для розробників розумних домашніх систем, оскільки вона забезпечує легкий доступ до мережевих ресурсів та віддалене управління.

Рисунок 1.8 – Зовнішній вигляд контролера «Raspberry Pi Zero»

Таблиця 1.2 – Основні технічні характеристики «Raspberry Pi Zero»

Характеристика	Опис
Центральний процесор	Одноядерний
Оперативна пам'ять	512 МБ
Підключення	Mini HDMI, USB On-The-Go порт
Підтримка Wi-Fi	Так
Підтримка Bluetooth	Так
GPIO піни	Для підключення датчиків та пристроїв
Мікро SD слот	Для зберігання даних та ОС
USB-з'єднання	Micro USB
Вихідне відео	Mini HDMI
Робоча напруга	3.3 В

Raspberry Pi Zero можна розглядати як ефективний інструмент для реалізації систем «Розумний дім». Його невеликі розміри та вбудовані можливості з'єднання, включаючи Wi-Fi та Bluetooth, роблять цю плату підходящою для інтеграції в різноманітні домашні середовища. Центральний одноядерний процесор та 512 МБ оперативної пам'яті забезпечують достатню обчислювальну потужність для

керування базовими домашніми автоматизаційними задачами. GPIO піни пропонують гнучкість у підключенні додаткових модулів та датчиків. Отже, Raspberry Pi Zero є практичним вибором для розробки та дослідження в області домашньої автоматизації.

STM32 Blue Pill

STM32 Blue Pill – це мікроконтролерна плата, яка знаходить широке застосування в сфері електронних проектів, від початкового рівня до більш складних систем (рис. 1.9). Вона ефективно використовується в системах «Розумний дім» для контролю та автоматизації домашніх процесів, завдяки своїй можливості інтеграції з різноманітними датчиками та пристроями. STM32 Blue Pill дозволяє створювати індивідуальні сценарії управління різними аспектами домашнього життя, такими як освітлення, клімат-контроль та системи безпеки, забезпечуючи гнучкість та ефективність управління домашнім середовищем.

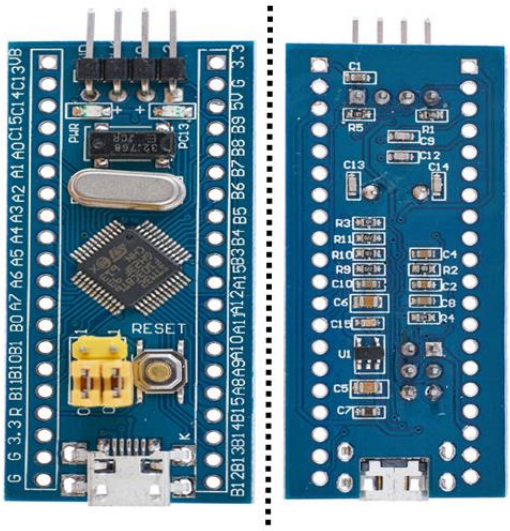


Рисунок 1.9 – Зовнішній вигляд контролера «STM32 Blue Pill»

Таблиця 1.3 – Основні технічні характеристики «STM32 Blue Pill»

Характеристика	Опис
Мікроконтролер	STM32F103C8T6 (ARM Cortex-M3)
Робоча напруга	3.3 В
Вхідна напруга	7-12 В (через регулятор напруги)
Цифрові входи/виходи	37 (всі GPIO піни)
Аналогові входи	10

Постійний струм на вході/виході	20 мА (на пін)
Частота кристала	72 МГц
USB-з'єднання	Micro USB
Пам'ять	64 КБ Flash, 20 КБ SRAM
Додаткові функції	CAN, I2C, USART, SPI, USB, ADC

STM32 Blue Pill є ефективним інструментом для використання у проектах систем «Розумний дім», завдяки своїй потужності, гнучкості та доступності. Ця мікроконтролерна плата, заснована на ARM Cortex-M3, вирізняється високою обчислювальною потужністю, розширеними можливостями підключення та великою кількістю входів/виходів. Ці характеристики роблять її перспективною для створення складних та високоінтегрованих рішень в рамках домашньої автоматизації, від керування освітленням і кліматом до систем безпеки та інших інтелектуальних додатків.

1.4 Сучасні методи забезпечення безпеки в системах «Розумний дім»

Системи безпеки тепер включають в себе широкий спектр інтелектуальних функцій, здатних адаптуватися до потреб та звичок мешканців. Штучний інтелект, зокрема машинне навчання та глибоке навчання, відкриває нові можливості для підвищення безпеки, дозволяючи системам не тільки реагувати на загрози, але й антиципувати їх. Аналізуючи методи ШІ у контексті «розумних будинків» допоможе зрозуміти, як інтелектуальні системи можуть захистити оселі та сприяти створенню безпечнішого та комфортнішого життєвого простору.

Методи домашньої автоматизації

Домашня автоматизація, базуючись на інтеграції систем безпеки, створює умови для глибокої взаємодії користувача з житловим середовищем. Ця технологія включає в себе різноманітні сенсори та актуатори, які реагують на команди від централізованої платформи управління. За допомогою інтернету речей, кожен пристрій у будинку може бути підключений до мережі, що дозволяє користувачам відстежувати стан своїх систем безпеки в будь-який час і з будь-якого місця [13].

Користувачі можуть налаштовувати сповіщення про вхід чи вихід, перевіряти статус дверних замків, відслідковувати відеозаписи з камер, а також керувати системами оповіщення та освітлення. Інтелектуальні замки надають можливість створювати тимчасові або постійні коди доступу для гостей чи сервісних служб, забезпечуючи при цьому високий рівень безпеки. Системи виявлення руху можуть інтегруватися з освітленням, щоб автоматично включати світло при виявленні активності, тим самим відлякуючи потенційних зловмисників.

Розумні будинки також можуть використовувати аналітику поведінки для оптимізації налаштувань безпеки, адаптуючи їх до звичок мешканців. Наприклад, якщо система помічає, що двері гаражу зазвичай відчинені в певний час дня, вона може нагадувати про закриття дверей або навіть автоматично їх закривати за певних умов. Ця інтеграція робить систему безпеки не тільки реактивною, але й превентивною.

Завдяки розвитку штучного інтелекту та машинного навчання, системи безпеки «розумних будинків» стають все більш просунутими, здатними вчасно реагувати на загрози та навіть передбачати потенційні інциденти, що робить житло не тільки інтелектуальним, але й дійсно «розумним» [14].

Методи розумного відеоспостереження

Розумне відеоспостереження відіграє революційну роль у системах безпеки «розумних будинків», використовуючи передові алгоритми аналізу відео для більш глибокого розуміння того, що відбувається в полі зору камер. Системи можуть виявляти не лише рух, але й розпізнавати особливості осіб, номерні знаки автомобілів, а також інші важливі атрибути. Інтеграція з штучним інтелектом дозволяє таким системам навчатися від попередніх інцидентів, покращуючи їх здатність прогнозувати потенційні загрози та забезпечувати проактивну реакцію.

За допомогою глибокого навчання, розумне відеоспостереження може аналізувати великі обсяги відеоданих, виявляючи складні шаблони поведінки та визначаючи, коли дії виходять за рамки звичного. Це включає в себе автоматичне відстеження та оповіщення про такі події, як несанкціонований доступ у заборонені

зони, відмінності у поведінці відвідувачів, чи навіть відстеження предметів, які були залишені або видалені з місця.

Крім того, інтеграція розумного відеоспостереження з домашніми автоматизаційними системами дозволяє створювати сценарії, в яких відеокамери співпрацюють з освітленням, замками, і навіть з аудіосистемами для створення повного відчуття присутності, що може відлякувати зловмисників. Ця технологія також може використовуватися для створення зручностей для мешканців, наприклад, для ідентифікації та привітання їх при приході додому чи для інформування про прибуття гостей.

Однією з найбільших переваг розумного відеоспостереження є його спроможність до самовдосконалення. Системи безпеки, які постійно навчаються на зібраних даних, стають краще з кожним використанням. Завдяки цьому, вони можуть передбачати потенційні безпекові інциденти, перш ніж вони стануться, і автоматично адаптувати свої реакції для максимального захисту дому та його мешканців [15].

Системи розумного освітлення

Системи розумного освітлення стали невід'ємною частиною інтелектуальної автоматизації будинку, вносячи значний вклад у безпеку та комфорт його мешканців. Ці системи здатні автоматично регулювати освітлення, враховуючи природнє світло, час доби та присутність людей в приміщенні. Вони можуть програмуватися на включення світла при виявленні руху, що не тільки забезпечує зручність для мешканців, але й діє як ефективний захід проти непроханих гостей, відлякуючи їх світлом.

Інтеграція з системами безпеки дозволяє освітленню реагувати на сигнали тривоги або інші безпекові повідомлення, включаючи світло для відбиття атаки або для підвищення якості зображення відеоспостереження. Це створює більш безпечне середовище та може бути частиною сценаріїв безпеки, наприклад, імітації присутності людей в будинку під час їх відсутності.

Розумне освітлення може бути також інтегроване з системами домашнього асистента для голосового керування, що додає ще один рівень зручності. Сучасні

системи освітлення також включають енергоефективні технології, такі як світлодіодні лампи, що зменшують електроспоживання та сприяють створенню екологічно сталого дому [16].

Інтегровані безпекові системи

Інтегровані безпекові системи, які базуються на принципах машинного навчання, представляють собою передове рішення для забезпечення безпеки в «розумних будинках». Вони аналізують поведінку користувачів, їхні повсякденні звички та рутини, вчаться розуміти «нормальний» стан речей і, як наслідок, можуть виявляти аномалії, що можуть вказувати на проблеми безпеки. Наприклад, якщо двері або вікно відчиняються в нетиповий час, система може відправити сповіщення власнику або активувати сигналізацію.

Ці системи використовують широкий спектр датчиків, включаючи датчики руху, камери, сенсори відкриття дверей/вікон, температури, диму та багато іншого, для збору даних про домашнє середовище. Машинне навчання дозволяє обробляти і аналізувати цю інформацію, виявляючи не тільки очевидні сигнали, але й здатне розпізнавати тонкі патерни поведінки або незначні зміни в звичному порядку речей.

Крім того, інтегровані системи можуть адаптуватися до змін у поведінці користувачів або до нових загроз, оскільки вони «вчаться» з часом, стаючи все більш точними в ідентифікації потенційних проблем. Це означає, що з часом вони вимагають менше ручного втручання і забезпечують більш комплексний рівень безпеки, який здатен пристосовуватись до еволюції власного середовища та його мешканців.

Такі системи забезпечують високий рівень персоналізації, дозволяючи налаштовувати параметри безпеки відповідно до особистих потреб і переваг. Це створює більш надійний і інтуїтивно зрозумілий шар захисту, який працює непомітно для користувачів, але завжди на варті, щоб забезпечити їхню безпеку.

Системи інтелектуальних замків безпеки

Інтелектуальні замки безпеки «розумного будинку» пропонують безключовий доступ і ряд передових функцій безпеки. Використання біометричних

технологій, таких як відбитки пальців, розпізнавання обличчя або сітківки ока, дозволяє точно ідентифікувати осіб, що намагаються отримати доступ до будинку, та запобігати несанкціонованому входу. Додатково, багато інтелектуальних замків можуть бути інтегровані з домашніми автоматизаційними системами і керовані через мобільні додатки, що надає власникам можливість контролювати доступ до їхнього дому незалежно від їхнього фізичного розташування.

Використання мобільних додатків також відкриває широкі можливості для управління доступом, включаючи створення тимчасових або постійних електронних ключів для гостей, сервісних служб або членів сім'ї. Ці ключі можуть бути швидко і легко скасовані або змінені, що забезпечує високий рівень гнучкості та контролю.

Деякі інтелектуальні замки також мають функції ведення журналу доступу, записуючи кожну спробу входу або виходу, що дозволяє власникам відстежувати активність навколо їхньої власності. Сучасні замки можуть також інтегруватися з іншими системами безпеки, такими як камери відеонагляду або системи розумного освітлення, щоб створювати більш комплексні безпекові сценарії.

Технології шифрування використовуються для захисту бездротового з'єднання між замком і мобільними пристроями, гарантуючи, що всі передані дані залишаються конфіденційними та захищеними від зловмисних спроб перехоплення. Крім того, інтелектуальні замки можуть підтримувати двофакторну автентифікацію, вимагаючи додаткового підтвердження, такого як код доступу або підтвердження на мобільному пристрої, перш ніж надати доступ.

Функціональність інтелектуальних замків значно розширює можливості традиційних замків, додаючи додатковий рівень безпеки та зручності. Вони можуть включати у себе підтримку Wi-Fi або Bluetooth, що дозволяє замкам синхронізуватися з домашньою мережею та надавати власникам повідомлення в реальному часі. Така інтеграція забезпечує моніторинг стану замків та управління ними з будь-якої точки світу. Автоматизація та персоналізація налаштувань замків дозволяють створити розклади автоматичного замикання чи відмикання дверей, а також активувати замки в залежності від геолокації користувачів. Ці та інші

розширені можливості інтелектуальних замків роблять їх незамінним елементом сучасних систем безпеки «розумного будинку» [17].

Методи інтеграції із персональними асистентами

Інтеграція систем безпеки «розумного будинку» з голосовими помічниками стала одним із найбільш прогресивних кроків в області домашньої автоматизації. Ця технологія надає користувачам можливість управління системами безпеки, як от сигналізації, відеонагляду, та контролю доступу через зручні голосові команди. За допомогою простих команд, користувачі можуть налаштовувати режими охорони, моніторити статус захисних систем, а також отримувати негайні повідомлення про будь-які безпекові інциденти в своєму домі.

Голосові помічники можуть інтегруватися з широким спектром пристроїв, включаючи дверні дзвінки з камерами, сенсори руху, димові детектори, та інші компоненти системи безпеки. Це дозволяє створювати сценарії, за яких домашній асистент може автоматично реагувати на різні події, такі як включення освітлення у випадку виявлення руху в нічний час або надсилання сповіщень, коли система фіксує незвичну активність.

Крім того, персональні асистенти можуть бути програмовані для взаємодії з декількома користувачами, розпізнаючи їх голоси та надаючи індивідуальні рівні доступу та контролю, що підвищує рівень персоналізації та безпеки. Ці системи також можуть використовувати штучний інтелект для навчання ваших звичок та автоматичного адаптування налаштувань безпеки для більш ефективної роботи.

Використання персональних асистентів для управління системами безпеки значно спрощує повсякденне життя, дозволяючи користувачам зосередитися на своїх справах, в той час як інтелектуальна система безпеки працює на захист їх дому.

Отже, методи домашньої автоматизації забезпечують централізоване управління безпекою дому через мобільні додатки. Розумне відеоспостереження використовує алгоритми аналізу відео для ідентифікації осіб та відхилень у поведінці. Системи розумного освітлення автоматично регулюють освітлення для підвищення безпеки та зручності. Інтегровані безпекові системи використовують

машинне навчання для моніторингу та реагування на незвичайну активність. Системи інтелектуальних замків пропонують безключовий доступ і дозволяють віддалене управління через мобільні додатки, забезпечуючи зручність та високий рівень захисту [18].

Окрім фізичного проникнення, іншою значною загрозою для систем «Розумний дім» є витоки інформації. Ці системи збирають та обробляють величезні обсяги даних про звички, переваги та поведінку користувачів, що робить їх привабливою мішенню для кіберзлочинців. Несанкціонований доступ до цих даних може призвести до порушення конфіденційності, а також бути використаним для інженерних атак або шахрайства. Важливим аспектом безпеки в «Розумному домі» є захист від витоків інформації, що включає в себе шифрування даних, захист мережевих з'єднань та регулярне оновлення безпекових протоколів. Це вимагає комплексного підходу до кібербезпеки, що об'єднує як апаратні, так і програмні рішення для забезпечення належного захисту даних.

1.5 Принципи роботи та класифікація мереж LSTM

LSTM (Long Short-Term Memory) — це спеціалізований тип рекурентних нейронних мереж (РНМ), який ефективно вирішує проблему зникнення градієнта, характерну для традиційних РНМ. LSTM спроектовані для запам'ятовування довготривалих залежностей та ефективної роботи з послідовностями даних, що робить їх ідеальними для завдань прогнозування у системах розумного дому.

На рис. 1.10 зображено алгоритм роботи мережі LSTM у вигляді діаграми активності.

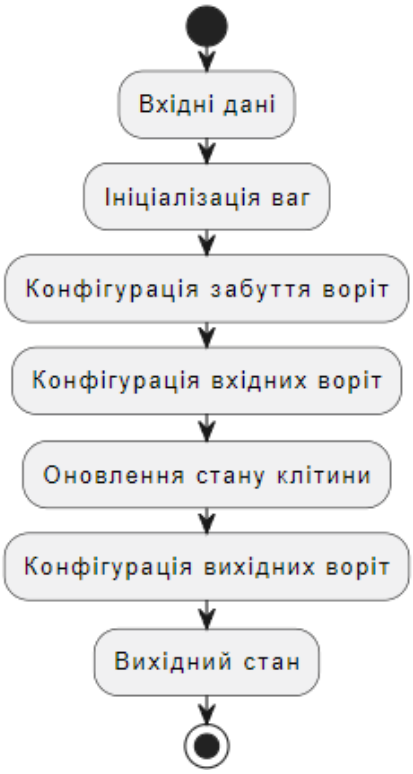


Рисунок 1.10 – Діаграма активності мережі LSTM

Алгоритм роботи LSTM розглядається як послідовність взаємопов'язаних кроків, кожен з яких відіграє важливу роль у збереженні та обробці інформації протягом часу. Вхідні дані спочатку подаються у мережу, де вони проходять через ряд воріт — структур, що вирішують, яку інформацію слід зберегти, оновити чи відкинути.

Ініціалізація ваг в мережі встановлює початкові параметри, які будуть адаптуватися під час навчання. Забуття воріт визначає, які частини існуючої інформації необхідно видалити з пам'яті клітини. Вхідні воріт вирішують, яку нову інформацію слід додати до стану клітини. Після цього відбувається оновлення самого стану клітини, яке враховує інформацію, отриману від забуття та вхідних воріт.

Вихідні воріт контролюють, яка частина оновленого стану клітини буде використана як вихідний сигнал. Цей вихідний сигнал може бути використаний як кінцевий результат або як вхід для наступного кроку в послідовності, дозволяючи LSTM передавати інформацію через час.

Ключовою особливістю LSTM є наявність так званих «вентилів» (gates) — структур, які регулюють потік інформації в мережі. Існує три основні типи вентилів у LSTM:

– вентиль забування (Forget Gate). Визначає, яку інформацію з попереднього стану варто «забути», щоб ефективно оновлювати стан комірки;

– вхідний вентиль (Input Gate). Вирішує, яка нова інформація буде зберігатися у стані комірки;

– вихідний вентиль (Output Gate). Визначає, яка частина стану комірки буде виведена на наступний шар або як вихід мережі.

Кожен з цих вентилів використовує власну нейронну мережу для вирішення, які дані потрібно пропустити через них. Така архітектура дозволяє LSTM зберігати інформацію на необмежений період часу, доки вона залишається актуальною для завдання, яке вирішує мережа.

Мережі LSTM вирішують складні задачі, пов'язані з аналізом послідовностей даних, надаючи можливість системам не тільки реагувати на поточні події, але й прогнозувати майбутні тенденції та потенційні загрози. Завдяки своїй спроможності зберігати інформацію протягом тривалого часу, LSTM мережі стають незамінними у розробці інтелектуальних систем, які вимагають високого рівня розуміння та адаптації до змінних обставин [19].

Залежно від специфічних потреб та завдань, мережі LSTM можна класифікувати на кілька основних типів:

– стандартні мережі LSTM є основою для розуміння концепцій LSTM і їхнього застосування. Вони містять послідовність LSTM-блоків, кожен з яких здатний зберігати інформацію на тривалий часовий проміжок. Ці мережі зазвичай використовуються для задач, де необхідно враховувати інформацію з попереднього контексту, наприклад, при обробці мови чи прогнозуванні часових рядів;

– двонаправлені LSTM розширюють стандартні LSTM, дозволяючи моделі збирати контекстну інформацію не тільки з минулого (як у стандартних LSTM), але й з майбутнього. Це досягається за допомогою двох окремих шарів LSTM, один з яких обробляє дані в прямому напрямку (від початку до кінця), а

інший — у зворотному (від кінця до початку). Цей тип моделі особливо корисний у задачах, де контекст з майбутнього може надавати важливу інформацію, наприклад, в розпізнаванні мови;

– глибокі LSTM структури включають кілька шарів LSTM, розташованих один за одним. Вихід кожного шару подається як вхід до наступного шару, дозволяючи моделі вивчати інформацію на різних рівнях абстракції. Ця модель може бути корисною в складних задачах, де необхідно виявляти тонкі візерунки в даних, наприклад, у складному прогнозуванні часових рядів або в автоматичній генерації музики.

Для підвищення захищеності системи розумного дому, LSTM можуть використовуватися для:

– прогнозування поведінки. LSTM можуть вивчати поведінкові патерни користувачів і прогнозувати майбутні дії, дозволяючи системі прогнозувати та попереджати небажані події;

– виявлення аномалій. Шляхом аналізу поведінки користувачів та використання пристроїв, LSTM може ідентифікувати відхилення від звичайних патернів, що може вказувати на вторгнення або інші загрози безпеці.

– оптимізація роботи пристроїв. Використовуючи прогнози поведінки користувачів, система може автоматично адаптувати роботу домашніх пристроїв для забезпечення безпеки та ефективності.

Використання LSTM у системах розумного дому відкриває широкі можливості для створення адаптивних, інтелектуальних систем, які можуть підвищити безпеку та комфорт житла [20].

1.6 Оцінка потенційних загроз та ризиків в системах «Розумний дім»

Навіть із високим рівнем технічного прогресу, інтелектуальні системи безпеки в «розумних будинках» стикаються із ризиками, пов'язаними з безпекою. Ці ризики можуть виникати через різноманітні вразливості, які, у свою чергу, можуть бути експлуатовані зловмисниками для отримання несанкціонованого доступу до систем управління будинком і, як наслідок, до чутливих персональних

даних користувачів. Аналізуючи потенційні загрози, важливо розглянути кілька ключових категорій:

Мережеві вразливості

Мережеві вразливості – це недоліки в мережевій інфраструктурі, які дозволяють несанкціонований доступ або атаки на систему. До таких вразливостей можна віднести:

– слабкі паролі. Нескладні або за замовчуванням паролі, які легко вгадати або знайти в Інтернеті, підвищують ризик взлому;

– незахищені Wi-Fi мережі. Відкриті або недостатньо захищені бездротові мережі можуть стати вразливими до атак «man-in-the-middle», де зловмисник може перехоплювати та модифікувати мережевий трафік;

– використання застарілих або вразливих протоколів. Застосування старих версій протоколів, таких як WEP для Wi-Fi, які містять відомі вразливості, збільшує шанси несанкціонованого доступу;

– відсутність шифрування. Незашифровані мережеві з'єднання дозволяють легко перехоплювати дані, що передаються між пристроями;

– фішинг та інші види соціальної інженерії. Це включає атаки, що спрямовані на отримання конфіденційної інформації від користувачів, такої як паролі або доступ до мережі;

– відсутність або погана реалізація мережевих захисних механізмів. Як-от брандмауери, системи виявлення та запобігання вторгнень, що можуть не виявити або заблокувати шкідливий трафік;

– програмне забезпечення маршрутизатора. Несвоєчасні оновлення або вразливості в прошивці маршрутизаторів можуть створити входи для атак;

– неадекватне управління патчами. Відсутність регулярного оновлення безпеки для мережевого обладнання та програмного забезпечення [21].

Фізичні вразливості

Фізичні вразливості – це слабкі точки в фізичному захисті системи, що можуть дозволити зловмисникам безпосередньо взламати або змінити фізичне обладнання. До таких вразливостей можна віднести:

– незабезпечені точки доступу. Це можуть бути вікна, двері або інші входи, які не обладнані належними засобами безпеки, наприклад, датчиками відкриття або руху;

– фізичний доступ до інфраструктури. Вразливості можуть виникнути, якщо зловмисник отримує доступ до важливих компонентів інфраструктури, таких як мережеві кабелі, маршрутизатори або сервери;

– незахищене обладнання. Безпекові камери, системи управління доступом або будь-які інші виконавчі пристрої, які легко доступні для фізичного маніпулювання, можуть стати мішенню для вандалізму або злому;

– відсутність або несправність замків і бар'єрів. Ненадійні або пошкоджені замки, а також відсутність бар'єрів, можуть дозволити проникнення у вразливі зони;

– фізичний доступ до інтерфейсів управління. Незахищені панелі управління або термінали, до яких можна отримати доступ без необхідності авторизації, відкривають шлях для несанкціонованих дій;

– неадекватне освітлення та відсутність спостереження. Темні або слабо освітлені ділянки навколо приміщення можуть ускладнити виявлення несанкціонованого доступу та дати зловмисникам перевагу;

– недостатній фізичний захист важливих компонентів. Системи безпеки повинні бути обладнані для витримування спроб знищення або несанкціонованого доступу;

– відсутність або слабка сигналізація. Недостатні або неефективні сигналізаційні системи можуть не зупинити зловмисників або не попередити власників про проникнення вчасно [22].

Програмні вразливості

Програмні вразливості – це слабкі місця або баги у програмному забезпеченні, що можуть бути експлуатовані для несанкціонованого доступу або втручання в роботу системи. Докладніше про ці вразливості:

– неправильна обробка даних. Програмне забезпечення може містити баги, які дозволяють зловмисникам вводити шкідливі дані, потенційно призводячи до збоїв системи чи виконання незапланованих команд;

– вразливості ін'єкції. Це стосується ситуацій, коли атакуючий може «ін'єктувати» шкідливий код у систему через вразливі точки введення, такі як веб-форми або інтерфейси API;

– застаріле програмне забезпечення. Системи, які не отримують регулярних оновлень, можуть містити відомі вразливості, які не були усунуті через відсутність патчів;

– несанкціоноване виконання коду. Якщо програмне забезпечення не належним чином обмежує виконання коду, зловмисники можуть запускати довільні команди, що може призвести до втрати контролю над системою;

– неправильне управління сесіями. Вразливості в управлінні сесіями можуть дозволити зловмисникам перехопити або відтворити сесії користувачів, отримуючи несанкціонований доступ до системи;

– вразливості на рівні конфігурації. Неправильні або незахищені налаштування систем можуть створити лазівки для втручання у роботу системи або доступу до чутливої інформації;

– відсутність шифрування. Відсутність або слабке шифрування даних, які передаються або зберігаються, може дозволити зловмисникам легко читати конфіденційну інформацію;

– неправильна обробка помилок. Якщо програмне забезпечення не обробляє помилки належним чином, це може надати зловмисникам інформацію, яку можна використати для подальших атак;

– відсутність принципу найменших привілеїв. Програмне забезпечення, яке надає користувачам або процесам більше прав, ніж необхідно, підвищує ризик шкідливих дій, якщо ці права будуть скомпрометовані [23].

Вразливості обладнання

Вразливості обладнання – це слабкі місця у фізичних компонентах системи, такі як застаріле або незахищене обладнання, що може бути скомпрометоване через

відомі технічні недоліки або через відсутність оновлень безпеки. Основні проблеми цих вразливостей:

– фізичний захист обладнання. Багато пристроїв Інтернету речей (IoT) не розраховані на фізичні атаки. Це означає, що фізичний доступ до такого обладнання може дозволити зловмисникам швидко порушити його працездатність або змінити його функціональність;

– застаріле обладнання. Старі версії апаратного забезпечення можуть містити вразливості, які вже відомі в кібербезпековій спільноті і для яких можуть існувати готові експлойти. Такі пристрої рідко отримують оновлення програмного забезпечення, що робить їх легкою мішенню для атак;

– відсутність оновлень безпеки. Навіть сучасні пристрої можуть бути вразливими, якщо виробники не надають регулярних оновлень безпеки або якщо користувачі не встановлюють їх своєчасно;

– інтегроване програмне забезпечення. Прошивка (firmware) та інше програмне забезпечення, вбудоване в пристрої, може містити помилки або вразливості, які зловмисники можуть використовувати для отримання контролю над обладнанням;

– використання стандартних паролів. Багато пристроїв використовують стандартні паролі, які легко знайти в інтернеті або з якими пристрої поставляються виробниками, і які користувачі часто не змінюють;

– незахищені порти та інтерфейси. Незахищені USB-порти, слоти для SD-карт або інші фізичні інтерфейси можуть дозволити несанкціонований доступ до пристроїв або встановлення шкідливого програмного забезпечення;

– відсутність шифрування. Обладнання, яке не використовує шифрування для зберігання або передачі даних, може дозволити зловмисникам легко перехопити та прочитати чутливу інформацію;

– побічні канали витоку інформації. Шкідливі процеси можуть використовувати побічні канали, такі як аналіз споживаної потужності або електромагнітне випромінювання, для витоку інформації з зашифрованих пристроїв.

Для забезпечення захисту, важливо систематично оцінювати ризики, що стосуються кожної з цих категорій, і застосовувати комплексні заходи безпеки. Це включає в себе регулярні оновлення програмного забезпечення, використання міцних паролів, шифрування даних, а також фізичні заходи безпеки для запобігання несанкціонованому доступу [24].

1.7 Огляд сучасних методів штучного інтелекту в системах безпеки

Сучасні ШІ-алгоритми здатні обробляти та аналізувати величезні обсяги даних з датчиків, камер та інших пристроїв, що дозволяє розпізнавати складні поведінкові шаблони і вчасно реагувати на потенційні загрози.

Методи машинного навчання дозволяють системам розпізнавати та реагувати на складні патерни поведінки. Класифікація дій користувачів полягає у визначенні типів поведінки, які можуть вказувати на звичайну діяльність чи потенційні загрози, такі як несанкціонований доступ або вандалізм.

За допомогою навчених моделей машинного навчання, системи можуть аналізувати інформацію з сенсорів або відеокамер і визначати, чи відповідає поведінка користувача встановленим шаблонам. Наприклад, якщо система знає, що користувач зазвичай повертається додому в певний час, відхилення від цього графіку може спричинити підвищену увагу системи [25].

Конволюційні нейронні мережі (КНМ) є одним з найпотужніших інструментів у сфері обробки відеоданих. Вони використовуються для розпізнавання образів та об'єктів у зображеннях та відео. В системах безпеки КНМ можуть ідентифікувати індивідуальні особи за їхніми особливостями обличчя або по фігурі. Це особливо корисно для систем контролю доступу, які використовують біометричні дані для верифікації осіб.

КНМ також можуть відстежувати переміщення об'єктів через послідовність кадрів, що дає змогу системі безпеки фіксувати траєкторії руху людей або транспортних засобів. Це важливо для виявлення підозрілих дій, як-от людина, яка зупиняється біля дверей кожного будинку у сусідстві.

Деякі розширені системи безпеки використовують алгоритми глибокого навчання для аналізу емоційних станів осіб на відео. Це може включати визначення ознак стресу, агресії або страху, що можуть бути попереджувальними сигналами про майбутній конфлікт або небезпечну поведінку.

Методи машинного навчання та глибокого навчання можуть інтегруватися з іншими системами розумного дому, такими як автоматизація освітлення та термостати, для створення цілісної безпекової системи. Наприклад, якщо система виявляє несанкціоноване вторгнення, вона може автоматично вмикати освітлення або блокувати доступ до будинку.

Використання ШІ в системах безпеки розумного дому не тільки збільшує ефективність та точність виявлення загроз, але й підвищує здатність системи до адаптації та прогностичного аналізу, забезпечуючи високий рівень безпеки для користувачів [26].

Також варто виділити і ансамблеві методи в машинному навчанні базуються на концепції, що об'єднання прогнозів з множини різноманітних моделей може забезпечити більшу точність і надійність, ніж використання однієї моделі. Вони важливі для систем безпеки, де помилкові позитиви або негативи можуть мати серйозні наслідки. Найбільш відомими ансамблевими методами є:

– випадкові ліси (Random Forests). Це метод, який створює множину дерев рішень, кожне з яких навчається на випадково відібраній підмножині навчального набору даних. Під час прогнозування кожне дерево в ансамблі голосує, і найпопулярніший вибір стає кінцевим прогнозом моделі. Це знижує варіативність та перенавчання, оскільки помилки від окремих дерев компенсуються іншими деревами в ансамблі; [27]

– градієнтний бустинг (Gradient Boosting). Градієнтний бустинг поступово створює ансамбль слабких моделей, зазвичай дерев рішень. Кожне нове дерево робить спробу виправити помилки попередніх, фокусуючись на тих випадках, де попередня модель виявилася неправильною. Ця стратегія дозволяє будувати модель, яка стає все сильнішою та точнішою з кожним новим деревом, що додається [28].

Перевагами ансамблевих методів є:

– зниження помилок варіативності. Як випадкові ліси, так і градієнтний бустинг дозволяють знизити помилки, пов'язані з випадковістю в навчальних даних;

– зменшення ризику перенавчання. Оскільки ансамблі використовують багато слабких моделей, вони менше схильні до перенавчання на шумі в даних;

– покращення стабільності. Ансамблеві моделі, як правило, демонструють більш стабільні прогнози у порівнянні з одиночними моделями, особливо при змінах у вхідних даних.

Отже, розвиток технологій Інтернету речей (IoT) відкриває нові перспективи для інтеграції ШІ в системи домашньої безпеки. Інтелектуалізація домашніх пристроїв трансформує їх зі статичних елементів у взаємопов'язані, інтелектуальні вузли, які активно спілкуються між собою та центральною системою управління.

IoT-пристрої, такі як датчики руху, камери, термостати, та освітлювальні системи, тепер можуть бути оснащені алгоритмами ШІ, що здатні аналізувати отримані дані в реальному часі. Вони вміють розпізнавати шаблони поведінки, передбачати потреби мешканців та виявляти аномалії, які можуть свідчити про безпекові інциденти.

Коли різноманітні IoT-пристрої інтегровані в єдину систему, вони можуть працювати синхронно, щоб забезпечити комплексний захист. Наприклад, якщо система виявляє незвичайну активність з допомогою датчиків руху, камери можуть активуватися для запису відео, а система освітлення може включитися для стримування потенційних зловмисників.

ШІ дозволяє системам безпеки розуміти звички та переваги користувачів, адаптуючи реакції відповідно до їхньої поведінки. Система може вчитися з часом, що підвищує її ефективність. Наприклад, якщо система знає, що мешканці зазвичай не вдома в певні години, будь-яка детектована активність протягом цього часу може бути визначена як підвищений ризик.

Індивідуальні налаштування безпеки стають можливими завдяки ШІ. Системи можуть враховувати особливості кожного мешканця, наприклад,

інвалідність або вік, для забезпечення належної безпеки. Це означає, що система безпеки може бути більш чутливою до допомоги літнім людям або менш чутливою до руху домашніх тварин.

Завдяки ШІ, системи безпеки можуть швидко адаптуватися до нових загроз або змін у домашньому середовищі. Вони можуть автоматично оновлювати свої алгоритми на основі зібраних даних та постійно вдосконалюватися для більш ефективного захисту.

Таким чином, інтеграція ШІ та IoT відкриває шлях до створення більш розумних, гнучких та пристосованих систем безпеки, які можуть забезпечити високий рівень захисту для розумних домівок [29].

1.8 Формулювання задачі та обґрунтування необхідності використання LSTM

Основним завданням є розробка алгоритму ідентифікації та нейтралізації кіберзагроз в реальному часі з використанням LSTM мереж, що впроваджуються у систему розумного дому. Алгоритм має бути реалізований на мові програмування C# із застосуванням бібліотеки ML .NET.

Важливою частиною інтеграції розроблюваного алгоритму ідентифікації та нейтралізації кіберзагроз є його впровадження у систему «Розумний дім». Це включає не тільки можливість виявлення та реагування на кіберзагрози, але й інтеграцію з іншими компонентами «розумного дому», такими як датчики руху, системи відеоспостереження, та інтелектуальне управління енергоспоживанням.

Алгоритм повинен бути сумісним з різними пристроями та платформами, що використовуються в «розумних домах», забезпечуючи таким чином гнучке та ефективне вирішення задач кібербезпеки. Наприклад, він може аналізувати дані з датчиків руху або камер, щоб ідентифікувати незвичайну активність, яка може бути ознакою спроби несанкціонованого доступу.

Також важливо, щоб система могла адаптуватися до індивідуальних потреб користувачів «розумного дому». Це означає можливість налаштування параметрів безпеки в залежності від конкретних умов та вимог, а також інтеграцію з різними користувацькими інтерфейсами для зручності кінцевих користувачів.

Завдяки цьому, розроблений алгоритм зможе ефективно виконувати свої функції в рамках комплексної системи «Розумний дім», забезпечуючи високий рівень безпеки та комфорту для користувачів.

Функціональні вимоги:

– детекція та ідентифікація загроз. Система має автоматично виявляти та ідентифікувати потенційні кіберзагрози та ненормативні поведінкові патерни, що могли б свідчити про кібератаку;

– прогнозування кібератак. З використанням LSTM мереж, система має прогнозувати можливі кібератаки, аналізуючи великі дані та поведінкові шаблони.

– реагування на інциденти. В разі виявлення загрози, система повинна виконувати автоматичне блокування атаки та повідомляти адміністратора системи;

– оновлення захисних механізмів. Система має підтримувати можливість оновлення захисних алгоритмів для відповіді на нові загрози;

– аудит та звітність. Запровадження функціоналу для збору та аналізу даних про безпекові інциденти, формування звітів.

Нефункціональні вимоги:

– продуктивність. Система має обробляти вхідні дані та реагувати на загрози в реальному часі без значних затримок;

– масштабованість. Система повинна бути спроектована таким чином, щоб її можна було масштабувати відповідно до зростаючих потреб користувачів та обсягів даних;

– надійність. Має бути забезпечена висока доступність системи і мінімальна кількість помилок у її роботі;

– безпека. Дані, що обробляються системою, повинні бути захищені від несанкціонованого доступу та витоку;

– інтегрованість. Система має легко інтегруватися з іншими компонентами системи «розумного дому»;

– зручність використання. Інтерфейс системи має бути інтуїтивно зрозумілим для кінцевого користувача.

Задля досягнення стратегічної мети побудови ефективного захисту «розумного дому», необхідність створення системи, що інтегрує в себе передові рішення в області інформаційної безпеки є важким завданням. Важливо приділити увагу не лише технічним характеристикам такої системи, а й її зручності для користувача, адже кінцевий успіх впровадження залежить від легкості використання та інтуїтивності інтерфейсів. Разом з тим важливо реалізувати надійність системи, яка повинна забезпечувати неперервну роботу в різноманітних умовах експлуатації, виявляючи та нейтралізуючи загрози без помилок чи збоїв.

1.9 Висновок до розділу

У рамках даного розділу було проведено аналіз і теоретичне обґрунтування концепції систем «Розумний дім», що дозволило виявити основні архітектурні рішення, які лягли в основу сучасних інтелектуальних будинків.

У процесі аналізу основна увага була приділена архітектурі та компонентам систем «Розумний дім», де було розглянуто схеми централізованих та децентралізованих підходів, а також визначено роль інтелектуальних систем в автоматизації домашніх процесів і забезпеченні безпеки. Було визначено, що застосування архітектур RL-IoT та AIoT є особливо перспективним у контексті вдосконалення реактивності та адаптивності системи.

Також було розглянуто методи забезпечення безпеки, починаючи від автоматизації домашніх процесів до розумного відеоспостереження, що дозволяє створювати багаторівневі захисні системи. Використання інтелектуальних замків, систем розумного освітлення, та інтеграція з персональними асистентами розкривають нові можливості для управління та моніторингу різноманітних сценаріїв безпеки.

Аналіз мереж LSTM та їх класифікація виявив значний потенціал цих технологій у покращенні інтелектуальних систем, зокрема у забезпеченні прогностичної аналітики та виявленні аномалій у поведінкових шаблонах, що є критичним для попередження та реагування на загрози.

Обґрунтування необхідності використання LSTM базується на їх здатності моделювати та аналізувати часові послідовності даних, що є характерними для багатьох процесів в домашньому господарстві. Така можливість дозволяє системі «Розумний дім» не просто реагувати на окремі події, а аналізувати тренди та прогнозувати майбутні сценарії, адаптуючись до змін у поведінці мешканців і внутрішньому середовищі.

2 ПРОЕКТУВАННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ БЕЗПЕКИ НА ОСНОВІ LSTM

2.1 Формалізація математичної моделі системи «Розумний дім» з використанням LSTM

Формалізація математичної моделі системи для підвищення захищеності розумного дому за допомогою ШІ передбачає створення моделі машинного навчання, яка може аналізувати та прогнозувати поведінкові патерни на основі даних, зібраних з датчиків розумного дому. Реалізація математичної моделі для системи безпеки розумного дому з використанням LSTM інкорпорує ряд послідовних кроків, які починаються від збору даних до виходу прогнозу. Описуючи цю модель, можна структурувати її у вигляді кількох математичних та алгоритмічних компонентів:

Модель LSTM

Математична модель LSTM (Long Short-Term Memory) складається з декількох ключових компонентів та етапів [30]:

2. Вхідні ворота (Input gate). Ці ворота вирішують, чи є вхідна інформація достатньо важливою, щоб бути збереженою. Формула:

$$i_t = \sigma(W^{(i)}x_t + U^{(i)}h_{t-1}) \qquad (2.1)$$

використовує сигмоїдну активаційну функцію для визначення, чи варто зберегти нову пам'ять $\widetilde{c_t}$.

2. Забувальні ворота (Forget gate). Визначають, чи потрібно забути інформацію з попередньої клітини пам'яті, за формулою:

$$f_t = \sigma(W^{(f)}x_t + U^{(f)}h_{t-1}) \qquad (2.2)$$

3. Вихідні ворота (Output gate). Вихідні ворота o_t вирішують, яку частину кінцевого стану клітини слід вивести як прихований стан. Використовуючи формулу:

$$o_t = \sigma\left(W^{(o)}x_t + U^{(o)}h_{t-1}\right), \qquad (2.3)$$

визначається, які частини кінцевого стану клітини c_t повинні бути використані для наступних кроків обчислень та передачі інформації до наступного шару мережі.

4. Генерація нової пам'яті. Нова пам'ять $\widetilde{c_t}$ створюється на основі поточного входу x_t та попереднього прихованого стану h_{t-1}. Формула:

$$\widetilde{c_t} = \tanh(W^{(c)}x_t + U^{(c)}h_{t-1})) \qquad (2.4)$$

дозволяє інтегрувати аспекти нового слова x_t у пам'ять.

5. Кінцева клітина пам'яті (Final memory cell). Кінцева клітина пам'яті c_t формується шляхом комбінування виходів забувальних воріт та вхідних воріт. Використовуючи формулу:

$$c_t = f_t \cdot c_{t-1} + i_t\widetilde{c_t} \qquad (2.5)$$

LSTM вирішує, яку частину інформації слід зберігати та яку - відкинути, забезпечуючи ефективне зберігання релевантної інформації для подальшого використання.

6. Прихований стан (Hidden state). Прихований стан h_t визначається за допомогою вихідних воріт та кінцевої клітини пам'яті. За формулою:

$$h_t = o_t \cdot \tanh(c_t) \qquad (2.6)$$

вихідні ворота вирішують, яка частина кінцевої пам'яті повинна бути виведена в якості прихованого стану, який є вихідним сигналом LSTM-блоку і використовується в наступних шарах мережі або для передбачення.

Отже, математична модель LSTM охоплює кілька ключових компонентів для ефективного оброблення та збереження інформації у часових рядах. Вона включає генерацію нової пам'яті, роботу вхідних, забувальних та вихідних воріт, а також механізми оновлення кінцевої клітини пам'яті та формування прихованого стану. Ці компоненти разом забезпечують здатність LSTM вибірково зберігати, оновлювати та видаляти інформацію, дозволяючи ефективно обробляти довгі залежності в даних та вирішувати проблему зникнення градієнта, типову для традиційних рекурентних нейронних мереж.

Вхідні дані

Вхідні дані для системи розумного дому включають декілька різновидів датчиків та сенсорів, які забезпечують інформацію про активності всередині житла. Для LSTM моделі, вхідні дані (X) представлені як набір векторів $X = \{X_1, X_2, \dots X_n\}$, де кожен вектор відповідає окремому інстансу події або спостереженню.

Кожен вектор містить наступні атрибути:

– StartTime: Час початку події, зазвичай у форматі часових міток;

– Location: Фізичне місцезнаходження події в будинку;

– TimeOfDay: Час доби, коли подія відбулася (ранок, день, вечір, ніч);

– Object: Об'єкт, з яким взаємодіяла особа (наприклад, двері, холодильник);

– Posture: Поза або стан особи під час події (сидячи, стоячи);

– Duration: Тривалість події у секундах;

– Activity: Опис активності, який використовується як мітка для класифікації.

Попередня обробка

Перед подачею вхідних даних у модель, необхідно їх підготувати через ряд етапів передобробки, щоб вони стали придатними для машинного навчання. Для цього потрібно провести:

– обробку часових міток (StartTime). Часові мітки можуть бути перетворені у числовий формат через вилучення годин, хвилин, та секунд як окремих атрибутів або через обчислення кількості секунд від певної початкової дати;

– кодування категорійних змінних (Location, TimeOfDay, Object, Posture, Activity). Для категорійних змінних використовується техніка One-Hot Encoding, що перетворює кожну категорію в бінарний вектор;

– нормалізацію числових змінних. Числові змінні, як-от тривалість події, слід нормалізувати для забезпечення однорідності масштабу з іншими атрибутами. Це може бути зроблено через віднімання середнього значення та поділ на стандартне відхилення (стандартизація) або масштабування до діапазону [0, 1] через мінімаксне масштабування;

– формування вхідних послідовностей. Після кодування та нормалізації, дані структуруються у послідовності, які приймаються LSTM.

Процеси передобробки забезпечують, що вхідні дані стають корисними для моделі, дозволяючи виявляти складні залежності та робити точні прогнози в поведінці користувачів та системних станів в рамках розумного дому.

Оцінка та валідація

Після тренування модель оцінюється за допомогою визначених метрик (точність, F1-міра, тощо), для перевірки її ефективності на тестових даних.

Розроблена математична модель буде використовватися для тренування моделі, яка може автоматично аналізувати поведінку в розумному домі та прогнозувати можливі вторгнення або інші надзвичайні ситуації, що підвищує безпеку та комфорт його мешканців.

2.2 Вибір та проектування архітектури

Вибір трьохрівневої архітектури для реалізації системи безпеки розумного дому можна обґрунтувати такими аспектами:

– модульність та гнучкість. Трьохрівнева архітектура дозволяє чітко розділити відповідальності, що сприяє легкості внесення змін, додавання нових функцій, тестування та підтримки системи;

– оптимізація обробки даних. Використання LSTM на рівні бізнес-логіки забезпечує ефективну обробку великих даних, зокрема для виявлення шаблонів поведінки та прогнозування;

– стійке зберігання та швидкий доступ до даних. Рівень даних забезпечує безпечне зберігання та швидкий доступ до інформації, що є критично важливим для системи безпеки розумного дому.

Обрана архітектура включає наступні компоненти:

– рівень користувацького інтерфейсу (Presentation Layer). Цей рівень становить границю між користувачем та системою. Його основне завдання - забезпечення інтуїтивного та зручного інтерфейсу для взаємодії з системою. Він включає в себе форми для тренування моделі, а також панелі управління та

моніторингу для перегляду результатів аналізу та втручання у роботу системи. Цей рівень також відповідає за представлення сповіщень та тривог, а також надання користувачам можливості керування налаштуваннями безпеки;

– рівень бізнес-логіки (Business Logic Layer). Рівень бізнес-логіки є ядром системи, де розміщені алгоритми машинного навчання і де здійснюється обробка та аналіз даних;[31]

– рівень даних (Data Layer). На рівні даних зберігаються всі необхідні вхідні та вихідні дані системи. Це включає бази даних для зберігання інформації про події, що відслідковуються, поведінкові патерни, результати тренування моделі та історію взаємодій користувачів з системою. Тут зберігаються також змодельовані LSTM моделі, які можуть бути використані для прогнозування у реальному часі [32].

Обрана трьохрівнева архітектура системи безпеки розумного дому забезпечить високий рівень модульності, що дозволяє незалежно розвивати та масштабувати кожен з рівнів. Це сприяє легкості внесення змін та додавання нових функцій, а також спрощує тестування та підтримку системи.

Розробка рівня даних застосунку є критичною складовою будь-якої системи, що оперує великими об'ємами інформації. У випадку системи безпеки для розумного дому, цей рівень відповідає за стійке зберігання, швидкий доступ та ефективне управління даними, зібраними з різноманітних датчиків.

Спочатку було створено шар доступу до даних, що представляє собою абстракцію, яка ізолює бізнес-логіку застосунку від безпосереднього доступу до бази даних. Це дозволяє змінювати структуру бази даних без впливу на інші частини системи та надає гнучкість для масштабування та підтримки. На рис. 2.1 представлено діаграму класів з методами для роботи з базою даних.

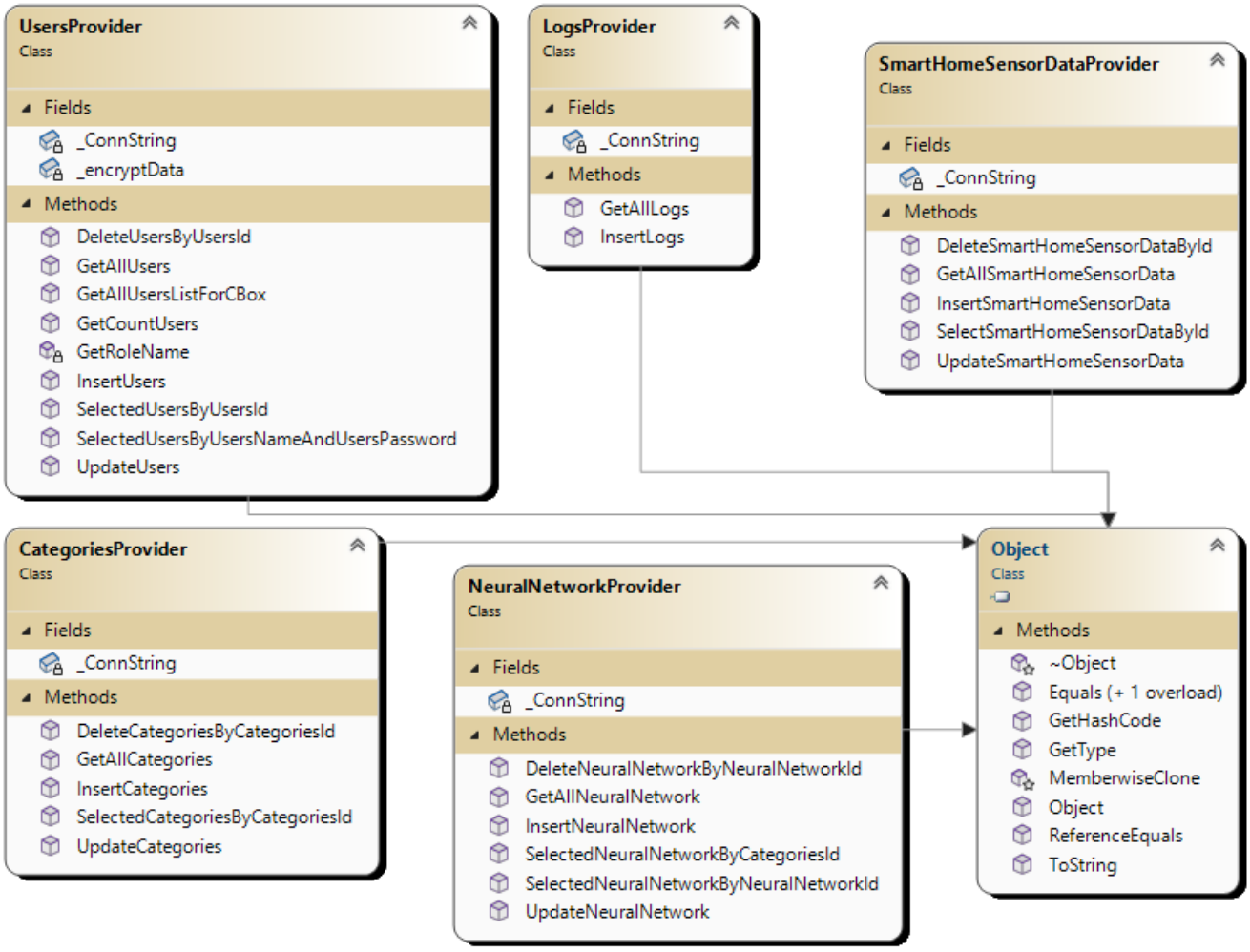


Рисунок 2.1 – Діаграма класів рівня даних

Як видно із рис. 2.1, діаграма класів рівня даних складається із 5-ти основних класів:

– клас CategoriesProvider є центральним для управління категоріями моделей LSTM у системі захисту. Він використовується для організації моделей за категоріями, що полегшує пошук та вибір конкретної моделі для задачі. Для кожної категорії можуть бути визначені унікальні атрибути та параметри, що дозволяють зберігати додаткову метаінформацію, як-от опис категорії чи пріоритет використання;

– клас LogsProvider веде реєстрацію всіх дій системи і взаємодій користувачів з системою. Він відіграє ключову роль у моніторингу стану системи та забезпечує відстеження змін конфігурації та потенційних безпекових подій. Журнали можуть бути використані для аудиту, діагностики проблем, та визначення шаблонів поведінки користувачів;

– клас NeuralNetworkProvider представляє собою фасад для взаємодії з нейронними мережами, що зберігаються в системі. Він дозволяє зберігати деталізовану інформацію про кожну з мереж, включно з її архітектурою, станом навчання, вагами, та історією тренувань. Можливість інтеграції зі зовнішніми обчислювальними ресурсами для тренування та оцінювання моделей також є частиною його функціоналу;

– клас SmartHomeSensorDataProvider збирає та управляє даними, отриманими від множини датчиків у розумному будинку. Він відповідає за агрегування, збереження та предоставлення доступу до даних, що є критично важливими для реального часу та історичного аналізу поведінки системи. Клас може обробляти великий об'єм даних та забезпечувати їх консолідацію для оптимізації подальшого аналізу;

– клас UsersProvider відповідає за управління користувацькими акаунтами в системі. Він забезпечує функції створення нових користувацьких профілів, аутентифікації, управління ролями та дозволами. Завдяки цьому класу, система здатна забезпечити індивідуалізований доступ до функцій та даних, базуючись на рівні прав користувача.

В архітектурі застосунку системи захисту на основі мережі LSTM було вирішено виділити інтерактивний рівень користувацького інтерфейсу. Розробка цього сегмента зосереджена на створенні інтуїтивно зрозумілих форм, які інтегрують різноманітні графічні елементи управління, такі як кнопки, текстові поля та таблиці для відображення даних. Ці компоненти забезпечують основу для двосторонньої взаємодії: вони дозволяють користувачам не тільки подавати інформацію, але й отримувати зворотній зв'язок та результати операцій.

Інтерфейс спроектований таким чином, щоб максимально спростити введення даних та навігацію по системі. Користувачі можуть швидко виконувати пошук потрібної інформації, запускати процеси аналізу та переглядати результати у формі, яка найкраще відповідає їхнім потребам і перевагам. Кожна взаємодія користувача з графічними елементами передає управління обробникам подій, які в

свою чергу комунікують з бізнес-логікою та рівнем даних, щоб забезпечити виконання запитів і повернення результатів (рис. 2.2).

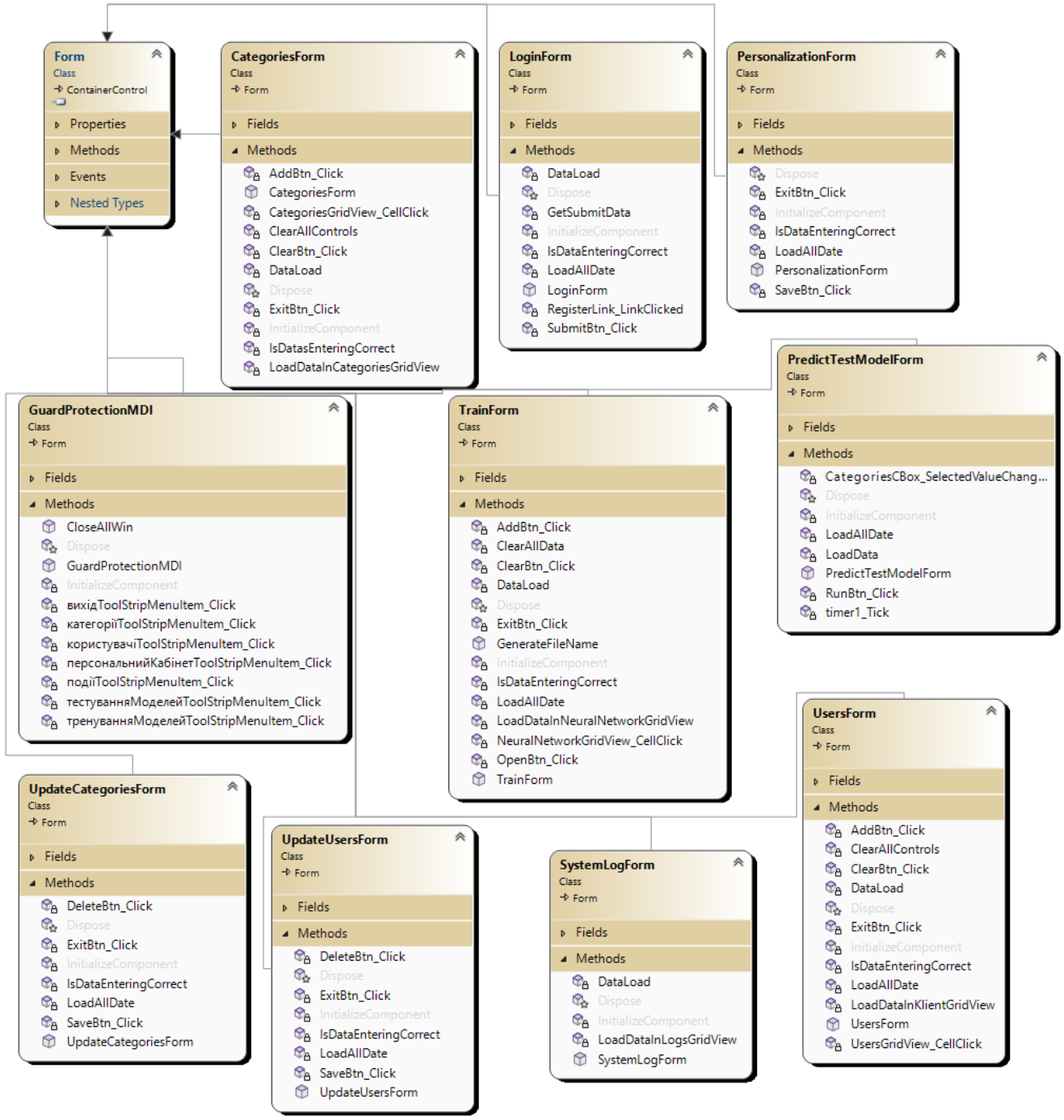


Рисунок 2.2 – Діаграма інтерфейсу системи

Діаграма даного рівня складається з десяти класів рівня UI та є похідними від класу Form, тобто мають графічний інтерфейс:

– GuardProtectionMDI. Це головне вікно додатку, яке служить як центральний хаб для навігації користувачів по системі. GuardProtectionMDI є багатодокументним інтерфейсом, що дозволяє одночасно відкривати та керувати

декількома вікнами в межах одного головного вікна. Кожне вікно в MDI може представляти різні аспекти системи, такі як тренування моделей, їх тестування чи управління категоріями.

– PredictTestModelForm. Це спеціалізована форма, яка дозволяє користувачам виконувати тестування різних моделей LSTM на симуляційних даних. Форма містить інтерфейс для введення тестових даних, запуску моделі та відображення результатів прогнозування. Це може бути корисним для перевірки ефективності моделей перед їхнім реальним застосуванням у робочій системі.

– CategoriesForm. Ця форма створена для управління категоріями моделей нейронних мереж. Вона надає інтерфейс для перегляду існуючих категорій, додавання нових, оновлення інформації про них та видалення непотрібних. Форма CategoriesForm полегшує організацію моделей та забезпечує легкість у їх пошуку та використанні.

– LoginForm - клас, що відповідає за інтерфейс входу в систему. Форма входу є першим вікном, з яким користувач взаємодіє, і вона забезпечує аутентифікацію користувачів за допомогою введення логіна та пароля. LoginForm може також включати функціональність відновлення пароля, реєстрації нового користувача та інтеграції з різними методами аутентифікації, такими як OAuth, двофакторна аутентифікація тощо.

– PersonalizationForm. Клас, що дозволяє користувачам налаштовувати різні аспекти системи згідно своїх персональних уподобань. PersonalizationForm може включати опції для зміни теми інтерфейсу, конфігурації сповіщень, налаштувань приватності, а також інших елементів користувацького інтерфейсу, які покращують індивідуальний користувацький досвід.

– Форма SystemLogForm є інструментом для перегляду журналів системи. Цей клас користувацького інтерфейсу використовується адміністраторами для моніторингу активності в системі, відстеження помилок, спроб входу та інших важливих подій. Форма може мати функції фільтрації та пошуку, щоб допомогти користувачам швидко знайти потрібну інформацію.

– UpdateUsersForm використовується для оновлення інформації про користувачів в системі. Це може включати зміну ролей користувачів, оновлення контактної інформації та інших персональних даних. Форма надає адміністраторам системи інструменти для керування обліковими записами користувачів, включаючи активацію, деактивацію та надання прав доступу до різних модулів системи.

– UsersForm є центральним місцем для управління користувачами системи. Ця форма дозволяє переглядати список усіх користувачів, додавати нових користувачів, видаляти існуючих та змінювати їх дані. Вона також може включати можливість перегляду та налаштування прав та ролей користувачів, що є важливим для забезпечення належного рівня безпеки та контролю доступу в системі.

Завершальним кроком у створенні архітектури застосунку стало розроблення бізнес-логіки, яка є фундаментом для всіх бізнес-процесів в межах системи. Ця складова архітектури втілює всі правила та алгоритми, які необхідні для забезпечення коректної бізнес-функціональності, і відіграє центральну роль у реалізації внутрішніх операцій додатку. Структурування бізнес-логіки в ізольовані блоки сприяє еластичності та масштабованості системи, а також її здатності адаптуватися до нових вимог без значного переписування коду.

Архітектура бізнес-логіки відокремлює бізнес-правила від інших шарів застосунку, таких як взаємодія з базою даних та презентаційний шар, що дозволяє розробникам зосередитися на логіці обробки даних, не замислюючись про технічні аспекти збереження даних чи їх представлення користувачам. Такий підхід не тільки поліпшує читабельність та підтримку коду, але й забезпечує легшу інтеграцію з різними інтерфейсами та базами даних, що робить систему більш гнучкою та відкритою для інновацій.

Ілюстрація цієї частини архітектури, представлена на діаграмі, чітко показує взаємодію між бізнес-логікою, виділяючи границі між ними та надаючи зрозумілий візуальний огляд структури системи (рис. 2.3).

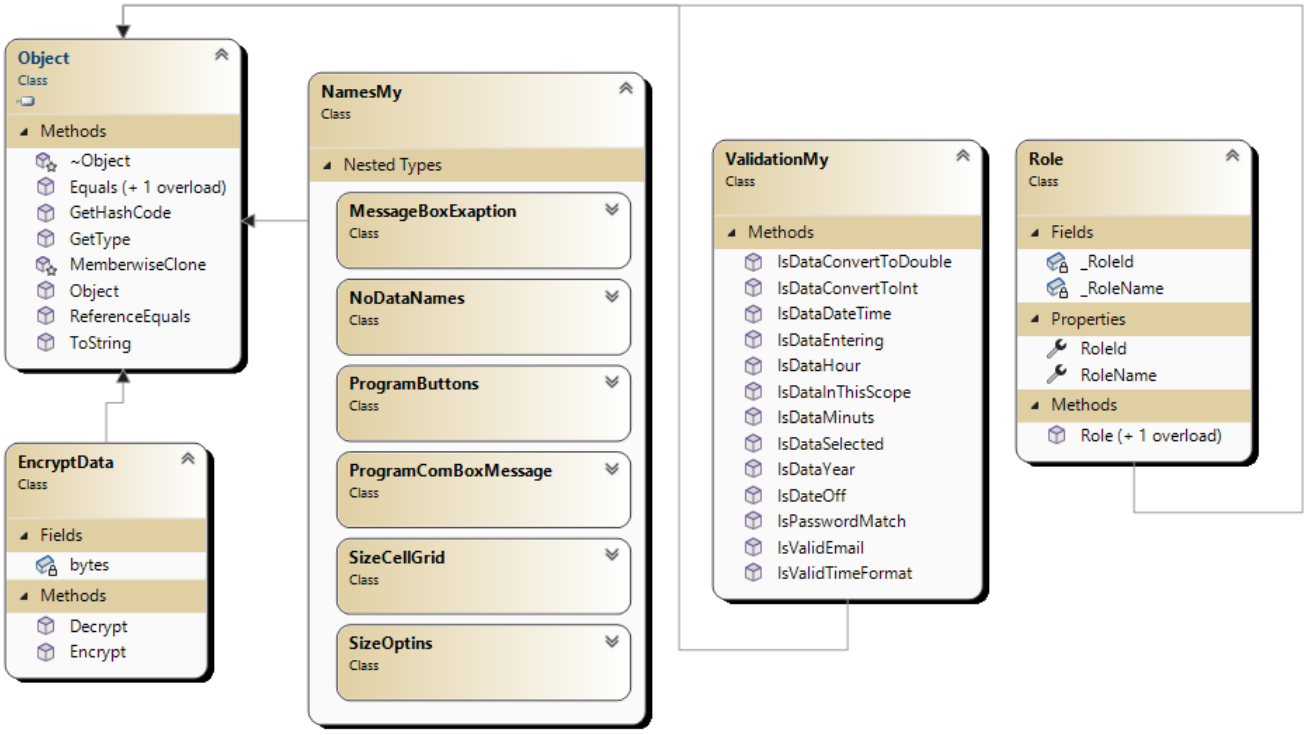


Рисунок 2.3 – Діаграма класів бізнес-логіки

У системі захисту на основі мережі LSTM, класи бізнес-логіки відіграють вирішальну роль у застосуванні бізнес-правил і алгоритмів, забезпечуючи безпеку, управління даними та валідацію. Основні класи бізнес-логіки складаються із:

– EncryptData. Даний клас відповідає за шифрування і розшифрування даних, які передаються або зберігаються в системі. Це може включати персональну інформацію користувачів, логи або навіть вихідні дані з датчиків. Завдяки використанню сучасних алгоритмів шифрування, EncryptData забезпечує, що чутливі дані залишаються захищеними від несанкціонованого доступу;

– NamesMy. Виступає як репозиторій для зберігання констант, які використовуються в програмі. Це може бути назви файлів, ключі налаштувань, ідентифікатори ресурсів та інші символьні константи. Використання такого класу сприяє легкості зміни значень, що часто використовуються, і підтримує чистоту коду, зменшуючи ймовірність помилок при введенні тексту;

– RoleApp. Відповідає за управління ролями в межах додатку. Цей клас визначає різні рівні доступу до функцій системи, що базуються на ролях користувачів. Він може використовуватись для перевірки прав доступу

користувачів під час їх взаємодії з системою та забезпечення, щоб користувачі могли виконувати лише ті дії, для яких вони мають дозвіл;

– ValidationMy. Клас ValidationMy є інструментом для перевірки валідності даних, які користувач вводить у систему. Він містить методи для перевірки форматів електронних адрес, телефонних номерів, паролів та інших вхідних даних. Застосування цього класу допомагає запобігти введенню некоректних даних в систему та гарантувати, що вся інформація, яка обробляється, відповідає очікуваним критеріям.

Кожен із розглянутих класів виконує важливу роль у забезпеченні надійності та стійкості системи захисту, створюючи надійний фундамент для бізнес-логіки, на якій будуються

2.3 Розробка алгоритмів для ідентифікації та класифікації потенційних загроз

Розробка алгоритмічної структури для інтелектуальних систем безпеки в контексті мереж LSTM розглядається як складний і багатогранний науково-дослідницький процес. На початку цього процесу лежить детальний збір даних з великої кількості датчиків, розміщених у домі. На рис. 2.4 показано блок-схему, яка демонструє процес навчання та зберігання моделі у системі.

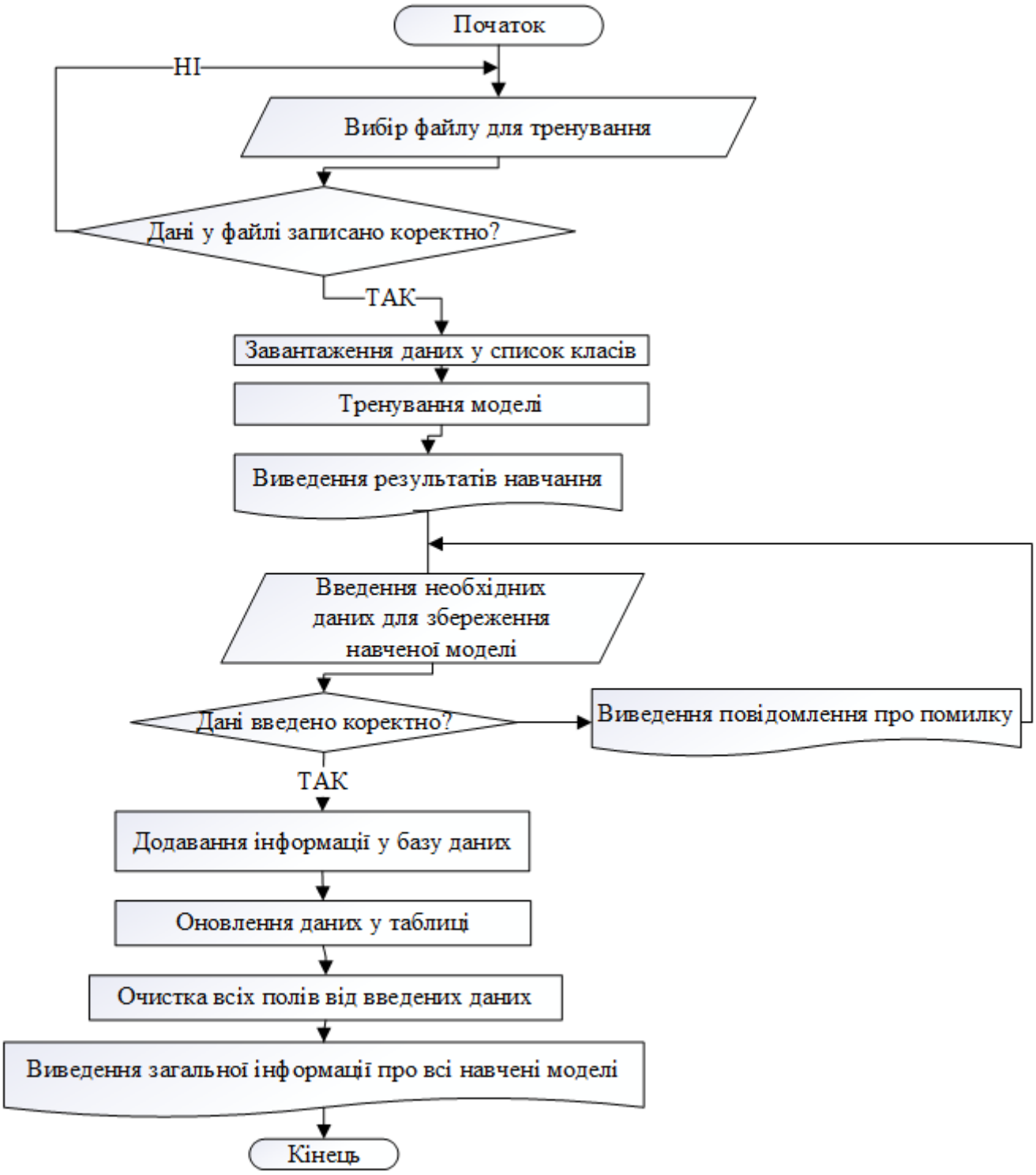


Рисунок 2.4 – Блок-схема алгоритму навчання та зберігання даних моделі

Процес навчання моделі ініціюється з вибору файлу, який містить необхідні тренувальні дані. Далі йде перевірка цих даних на коректність, після чого вони завантажуються для подальшого використання. Модель машинного навчання тренується на основі цих даних, а результати тренування виводяться для оцінки. Для збереження навченої моделі вводяться відповідні дані, які після перевірки на

правильність, додаються до бази даних. Після цього оновлюються дані в таблиці, що відображає інформацію про моделі, і поля вводу очищаються для наступного циклу тренування. В кінці виводиться інформація про всі навчені моделі.

Рис. 2.5 відображає блок-схему алгоритму роботи навченої моделі у реальному часі.

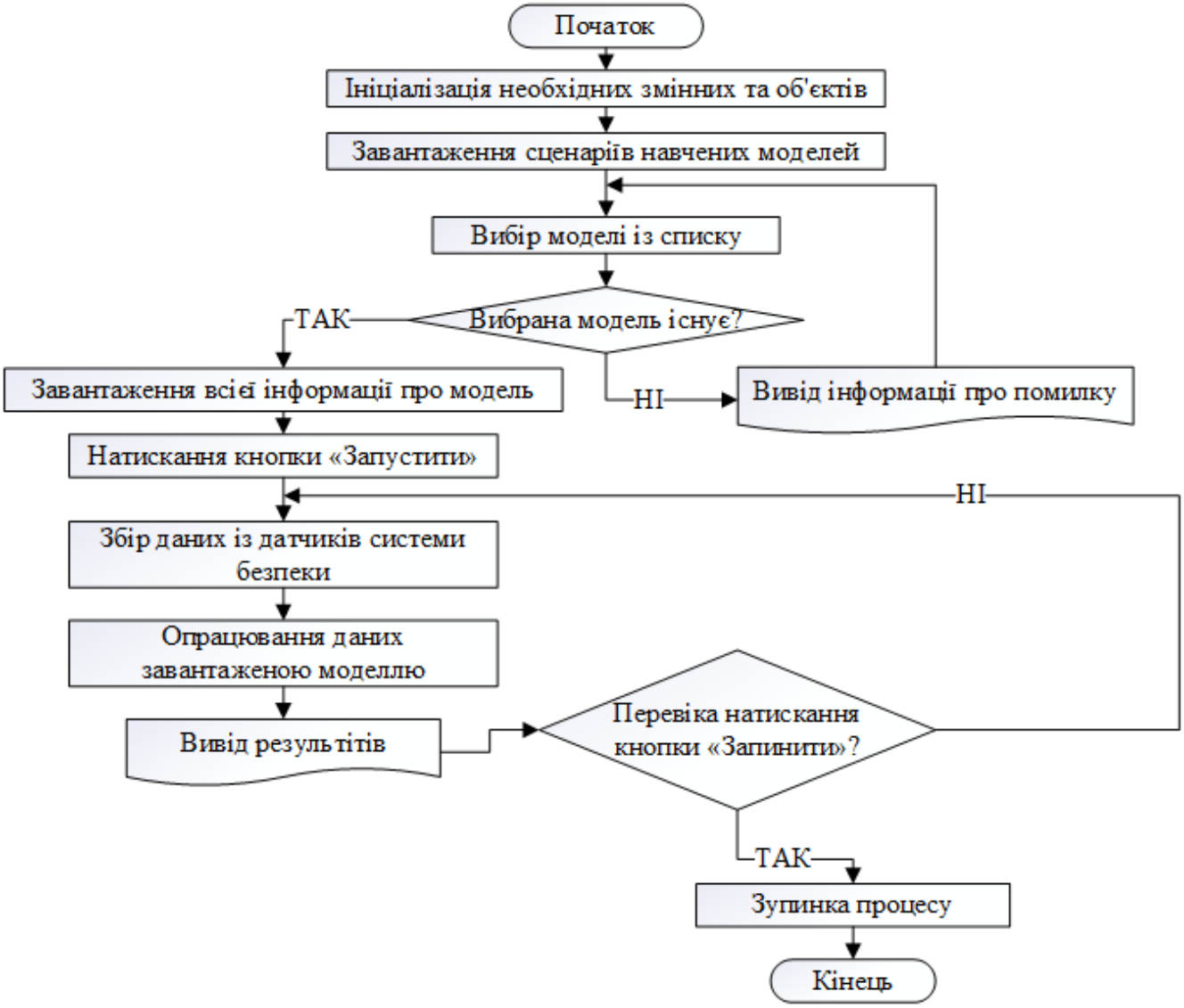


Рисунок 2.5 – Блок-схема алгоритму роботи моделі

Алгоритм використання навченої моделі для обробки даних в реальному часі включає кілька етапів. Спочатку ініціалізуються всі необхідні змінні та об'єкти. Потім завантажуються сценарії навчених моделей, після чого користувач вибирає потрібну модель зі списку. Якщо модель доступна, завантажується її інформація. Після натискання кнопки «Запустити» відбувається збір даних з датчиків системи безпеки та їх подальша обробка вибраною моделлю. Результати опрацювання

виводяться користувачу. Процес може бути зупинений за допомогою кнопки «Зупинити».

2.4 Вибір та обґрунтування комплектуючих для системи «Розумний дім»

2.4.1 Критерії вибору контролера

Для імплементації інтелектуальної системи безпеки в розумному будинку було обрано Arduino UNO. Даний контролер є втіленням принципів відкритої архітектури, що забезпечує розробникам гнучкість у підключенні широкого спектру сенсорів для збору даних, від датчиків температури до розширених систем відеоспостереження. Його легкість у налаштуванні та програмуванні робить Arduino UNO ідеальним вибором для швидкого прототипування і валідації концептів безпеки розумного дому.

Arduino UNO поєднує в собі надійність та стабільність, що є надзвичайно важливим для систем, які повинні працювати цілодобово без перебоїв. Його програмне забезпечення та апаратні можливості дозволяють ефективно обробляти сигнали з різних джерел та управляти даними в реальному часі, що є основою для аналізу даних LSTM моделями.

Популярність Arduino UNO забезпечила створення великої екосистеми модулів та бібліотек, які можуть бути легко інтегровані в будь-який проект. Це знижує технічний поріг входження та відкриває можливості для інновацій та творчості у сфері домашньої автоматизації та безпеки. Крім того, платформа відмінно підходить для освітніх цілей, надаючи студентам та дослідникам могутній інструмент для навчання та експериментів у сфері Інтернету речей та штучного інтелекту.

При виборі контролера для системи розумного дому було враховано ряд критичних критеріїв, зокрема:

– універсальність. Arduino UNO є однією з найбільш популярних платформ для прототипування, завдяки своїй гнучкості та широкому спектру можливостей взаємодії з різноманітними датчиками та модулями. Це робить його ідеальним для експериментування та розробки систем різної складності;

– доступність та вартість. Arduino UNO відомий своєю доступністю та вигідним співвідношенням ціни до функціональності. Це дозволяє зменшити загальні витрати на розробку системи, роблячи експериментування та розгортання більш економічно ефективними;

– простота програмування. Arduino UNO пропонує простий у використанні інтегрований розвиток середовища (IDE), що підтримує мову програмування C/C++. Це робить платформу доступною для розробників всіх рівнів кваліфікації;

– апаратна сумісність. Arduino UNO має стандартизований форм-фактор та велику кількість входів/виходів, що дозволяє легко підключати широкий спектр датчиків та модулів, необхідних для збору даних системи безпеки розумного дому;

– масштабованість. Хоча Arduino UNO може мати обмеження з точки зору обчислювальної потужності, його легко інтегрувати з більш потужними обчислювальними системами або іншими Arduino платами для створення масштабованих рішень;

– надійність. Arduino UNO вже довгий час є стандартом для хобістів та професіоналів, зарекомендувавши себе як надійна платформа, що може безперервно працювати в різних умовах, що є критично для систем безпеки.

Обрання Arduino UNO для збору інформації в системі розумного дому на основі LSTM враховує не лише технічні можливості плати, але й зовнішні фактори, такі як підтримка спільноти, вартість розробки та швидкість реалізації проекту [34].

2.4.2 Критерії вибору основних додаткових компонентів системи

При створенні системи безпеки для розумного будинку ключову роль відіграє вибір апаратної складової. Цей вибір ґрунтується на потребі використання надійних, високоякісних та точних компонентів, здатних забезпечити ефективну роботу системи безпеки. Важливі компоненти, такі як датчики руху, температурні датчики, мікрофонні модулі, та інші, відіграють ключову роль у зборі даних та відповіді на зміни умов оточуючого середовища.

Кожен компонент у цьому наборі обирається з огляду на його технічні характеристики та внесок у загальну систему безпеки. Наприклад, датчики руху використовуються для виявлення несанкціонованого проникнення, тоді як температурні датчики можуть відстежувати різкі зміни температури, які можуть бути ознаками пожежі або інших надзвичайних ситуацій. Мікрофонні модулі можуть виявляти підозрілі звуки, що також є важливим для систем безпеки.

Таким чином, кожен компонент у цьому вибраному наборі відіграє визначену роль у системі інтелектуальної безпеки, при цьому їх вибір базується на вимогах до точності, надійності, та сумісності з іншими елементами системи. Це забезпечує не тільки високий рівень безпеки, але й гарантує ефективність і стабільність роботи системи «Розумний дім» в цілому.

Датчик pir-d203s

Використання датчиків руху є важливим для реалізації функцій систем безпеки, та інших автоматизованих функцій. У табл. 2.1 приведено порівняльний аналіз датчиків руху.

Таблиця 2.1 – Порівняльний аналіз датчиків руху

Характеристика	PIR-D203S	HC-SR501	AM312
Дальність детекції	до 5 м	до 7 м	до 3 м
Кути детектування	120°	110°	100°
Напруга живлення	3-6V	4.5-20V	2.7-12V
Споживана потужність	низька	висока	помірна
Чутливість	висока	середня	висока
Час затримки	налаштовується	налаштовується	2с
Розміри	компактні	стандартні	компактні

Обираючи PIR-D203S для системи «Розумний дім» можна отримати перевагу у формі ширшого кута зору (120°), що дозволяє охоплювати більшу площу без необхідності встановлення додаткових датчиків. Хоча його дальність детекції складає 5 метрів, цього достатньо для більшості житлових та комерційних застосувань. Низька споживана потужність та можливість налаштування часу затримки роблять цей датчик енергоефективним та гнучким для різних сценаріїв

використання, що є важливим для «Розумного дому». Датчик PIR (Passive Infrared) моделі D203S є електронним пристроєм, що використовується для виявлення руху в просторі, зокрема людського присутності, за допомогою інфрачервоного випромінювання (рис. 2.6). Ці датчики широко використовуються в системах безпеки та автоматизації домівок через їхню здатність реагувати на зміни у випромінюванні, що відбуваються при переміщенні осіб або об'єктів з тепловим випромінюванням, яке відрізняється від температури навколишнього середовища.

Рисунок 2.6– Зовнішній вигляд датчика руху D203S

Датчик D203S містить два елементи, які вимірюють інфрачервоне випромінювання. Коли людина чи тварина переміщуються в зоні детекції датчика, випромінювання, що відбивається від цих об'єктів, змінюється, що викликає електричний сигнал у датчику. Цей сигнал може бути подальше оброблено контролером або мікропроцесором для активації певних дій, як-от вмикання світла, сповіщення системи безпеки або запису відео.

У контексті системи безпеки розумного дому з використанням LSTM, датчик PIR D203S буде відігравати роль вхідного пристрою для збору даних, які подальше аналізуються LSTM-мережею для визначення звичних моделей поведінки та виявлення аномалій. Наприклад, якщо система навчена розпізнавати звичайні часові рамки активності в будинку, спрацьовування датчика PIR у незвичний час може бути класифіковано як потенційна загроза та викликати відповідну реакцію системи.

Температурний датчик DHT22

Температурні датчики є важливими компонентами в багатьох системах контролю та автоматизації, зокрема в проектах «Розумний дім» та інших IoT-застосуваннях. Особливо важливою є точність і надійність таких датчиків, що

забезпечує правильну роботу системи. DHT22 вважається одним з найпопулярніших датчиків, завдяки його високій точності та надійності. У табл. 2.1 представлено порівняльні характеристики датчика DHT22 порівняно з іншими популярними датчиками, як DHT11 і DS18B20, щоб зрозуміти, чому DHT22 може бути кращим вибором для певних застосувань.

Таблиця 2.2 – Порівняльний аналіз температурних датчиків

Характеристика	DHT22	DHT11	DS18B20
Діапазон вимірювання температури	-40°C до +80°C	0°C до 50°C	-55°C до +80°C
Точність вимірювання температури	±0.5°C	±2°C	±0.5°C
Діапазон вимірювання вологості	0% до 100%	5 - 95% RH	Не застосовується
Точність вимірювання вологості	±2% RH	±5% RH	Не застосовується
Інтерфейс	Цифровий (1-Wire)	Цифровий (1-Wire)	Цифровий (1-Wire)
Частота оновлення даних	2 секунди	1 секунда	750 мс

Отже, DHT22 переважає DHT11 за такими параметрами, як ширший діапазон вимірювання температури та вологості, а також вища точність вимірювання. Це робить його більш універсальним для різних умов експлуатації. На відміну від DS18B20, DHT22 також вимірює вологість, що робить його більш підходящим для застосувань, де необхідно вимірювати обидва ці параметри.

Температурний датчик DHT22 складається з ємнісного датчика вологості та термістора, а також містить в собі АЦП для перетворення аналогових значень вологості та температури (рис. 2.7).

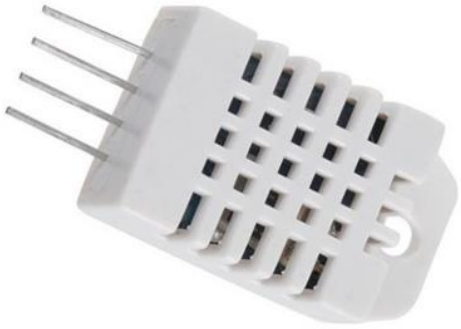


Рисунок 2.7– Зовнішній вигляд температурного датчика DHT22

DHT22 здатен вимірювати температуру з діапазоном від -40 до +80 градусів за Цельсієм із точністю до +/- 0.5°C, а також відносну вологість повітря в діапазоні від 0 до 100% з точністю до +/- 2% [36]. Датчик має цифровий вихід, що робить його зручним для використання з мікроконтролерами, такими як Arduino, без необхідності додаткових компонентів для аналого-цифрового перетворення сигналу.

У системі розумного дому, що використовує LSTM мережі для забезпечення безпеки, DHT22 буде виконувати моніторинг клімату.

Температурний магнітний датчик Reed module KY-025

Магнітні датчики використовуються у різноманітних застосуваннях, від простих домашніх проектів до складних промислових систем. У табл. 2.3 проведено порівняльний аналіз різноманітних датчиків, які можна використати для розробки системи розумного дому.

Таблиця 2.3 – Порівняльний аналіз магнітних датчиків

Характеристика	Reed Module KY-025	Hall Effect Sensor (A3144)	Magnetoresistive Sensor (HMC5883L)
Напруга живлення	3.3V - 5.5V	4.5V - 24V	2.16V - 3.6V
Розміри плати	1.5cm x 3.6cm	Залежить від моделі	Залежить від моделі
Вихідний сигнал	Цифровий	Цифровий	Аналоговий
Додаткові компоненти	LM393 компаратор	Немає	Немає
Чутливість до сильних магнітних полів	Висока	Середня	Середня

Як можна побачити із табл. 2.3 Reed Module KY-025 вирізняється своєю спрощеною інтеграцією завдяки вбудованому LM393 компаратору та високою чутливістю до сильних магнітних полів, що робить його відмінним вибором для проектів, де потрібна надійність і простота використання.

Магнітний датчик Reed module KY-025 є компонентом, який виявляє магнітне поле і змінює свій стан з відкритого на замкнутий або навпаки, коли поруч знаходиться магніт (рис. 2.8). Цей модуль складається з маленького скляного корпусу, в якому знаходяться два металеві контакти та магніт, який змушує ці контакти замикатися або розмикатися.

Рисунок 2.8– Зовнішній вигляд датчика Reed module KY-025

У системі захисту розумного будинку, Reed module KY-025 буде використовуватися для створення простої, але дуже ефективної системи виявлення стану відкриття або закриття дверей та вікон. Коли магніт, прикріплений до рухомої частини двері або вікна, наближається до датчика, який встановлений на рамі, контакти замикаються, що вказує на закритий стан [37]. Якщо двері чи вікно відчиняються, магніт віддаляється, контакти розмикаються, і цей стан фіксується системою.

Інтеграція таких датчиків у систему захисту дає змогу автоматично реагувати на несанкціонований вхід, наприклад, сповіщаючи господарів або активуючи сигнал тривоги. Завдяки своїй простоті та надійності, Reed module KY-025 є цінним доповненням до комплексної системи домашньої автоматизації, дозволяючи не тільки підвищити безпеку, але й втілювати різноманітні автоматизовані сценарії, такі як керування освітленням або системами клімат-контролю.

Окрім того, використання Reed module KY-025 дозволяє отримувати важливі дані для системи, яка будуть аналізувати поведінкові моделі користувачів і відповідно пристосовувати роботу інших систем безпеки в домі, забезпечуючи розумний і адаптивний захист.

Фоторезистор KY-018

Фоторезистори є фундаментальними компонентами в багатьох електронних проектах, особливо коли мова йде про вимірювання рівнів освітленості. Вони знаходять широке застосування в різних проектах, від простих освітлювальних схем до складних систем автоматизації. У табл. 2.4 проведено порівняльний аналіз найбільш підходящих фоторезисторів для реалізації проекту.

Таблиця 2.4 – Порівняльний аналіз фоторезисторів

Характеристика	KY-018	GM5528	GL5516
Напруга живлення	3.3V - 5V	Залежить від підключення	Залежить від підключення
Вихідний сигнал	Аналоговий	Аналоговий	Аналоговий
Тіньовий опір	500 кОм	1.0 МОм	500 кОм
Розміри	Компактні	Залежать від моделі	Залежать від моделі
Чутливість до світла	Висока	Висока	Висока
Робоча температура	-40°C до +85°C	-30°C до +70°C	-30°C до +70°C
Придатність для Arduino	Висока (готовий модуль)	Потребує додаткового налаштування	Потребує додаткового налаштування

Вибір фоторезистора KY-018 як компонента для проектів Arduino може бути обґрунтований на основі декількох ключових характеристик, виявлених у порівняльній табл. 2.4. Перш за все, KY-018 працює в діапазоні напруги 3.3V до 5V, що є стандартним для більшості Arduino-плат. Це робить його сумісним з широким спектром контролерів без потреби в додаткових регуляторах напруги.

Тіньовий опір KY-018 становить 500 кОм, що забезпечує хороший баланс чутливості для загальних застосувань. Це робить його більш універсальним порівняно з GM5528, який має тіньовий опір 1 МОм, що може бути надто високим для деяких застосувань. Компактні розміри KY-018 роблять його ідеальним для проектів, де простір є обмеженим, а висока чутливість до світла забезпечує точні вимірювання навіть у умовах слабкого освітлення. Робочий діапазон температур KY-018 (-40°C до +85°C) є ширшим порівняно з GM5528 та GL5516, що робить його більш стійким у екстремальних умовах. Також, KY-018 є готовим модулем

для Arduino, що зменшує складність інтеграції та забезпечує високу придатність для широкого спектра проектів, на відміну від GM5528 і GL5516, які вимагають додаткового налаштування. Це робить KY-018 відмінним вибором для користувачів різного рівня досвіду, особливо для тих, хто шукає простоту використання.

Фоторезистор KY-018 (рис. 2.9) є чутливим до світла компонентом, який змінює свій електричний опір в залежності від інтенсивності освітлення, що на нього падає [38]. Ця зміна опору може бути використана для активації або деактивації електронних схем у відповідь на зміну освітленості навколишнього середовища. У системах захисту розумного будинку фоторезистори часто використовуються для виявлення присутності або відсутності світла, що може вказувати на присутність людей або на автоматичне управління освітленням.

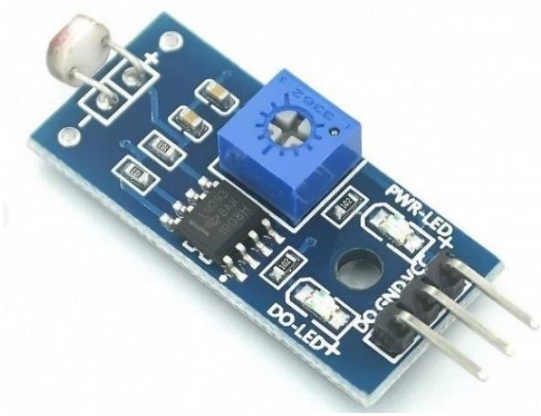


Рисунок 2.9– Зовнішній вигляд фоторезистора KY-018

В системі для аналізу поведінкових патернів, дані з фоторезистора KY-018 бути використані для навчання моделі розпізнавати звичайні та аномальні шаблони освітленості, що дозволить системі більш точно ідентифікувати потенційні загрози чи необхідність у втручанні.

Мікрофонний модуль MAX9814

Мікрофонні модулі також є важливим компонентом для даного проекту. Вибір правильного мікрофонного модуля може суттєво вплинути на якість звуку та загальну функціональність системи. У табл. 2.5 розглянуто три популярних мікрофонних модулі - MAX9814, LM4881 та MAX4466, що дасть змогу зрозуміти,

які їхні унікальні характеристики роблять один із них більш вигідним для певних застосувань.

Таблиця 2.5– Порівняльний аналіз мікрофонних модулів

Характеристика	MAX9814	LM4881	MAX4466
Напруга живлення	2,7 В - 5,5 В	2.7 В до 5.5 В	2.4 - 5.5 В
Тип підсилювача	Мікрофонний підсилювач	Мікрофонний підсилювач	Мікрофонний підсилювач
Автоматичне регулювання посилення (AGC)	Є	Немає	Немає
Налаштування посилення	40dB, 50dB, 60dB	Змінне	Змінне
Вихідний сигнал	Аналоговий	Аналоговий	Аналоговий

У даному випадку обрано модуль MAX9814 так як основні його переваги включають вбудоване автоматичне регулювання посилення (AGC), що дозволяє підтримувати стабільний рівень аудіосигналу незалежно від відстані до джерела звуку. Також модуль пропонує три рівні посилення (40dB, 50dB, 60dB), що робить його більш гнучким для різних аудіозастосувань. Низький рівень шуму цього модуля також є важливою перевагою для якісного запису звуку.

Отже, мікрофонний модуль MAX9814 (рис. 2.10) є високоякісним аудіопідсилювачем з автоматичним контролем виграшу, призначений для застосування у великому діапазоні звукових систем [39]. Цей модуль включає в себе мікрофон на електретній основі, який здатен вловлювати звукові хвилі, та інтегрований підсилювач для збільшення амплітуди звукового сигналу.

Рисунок 2.10– Зовнішній вигляд модуля MAX9814

Однією з ключових особливостей MAX9814 є його можливість автоматичного регулювання виграшу, що дозволяє модулю адаптуватися до різних рівнів гучності в середовищі без втручання користувача.

У системах захисту розумного будинку, MAX9814 буде виявляти акустичні події, які можуть вказувати на потенційні загрози або надзвичайні ситуації [40]. Завдяки високій чутливості та здатності до збору якісного звукового сигналу, модуль може використовуватися для розпізнавання специфічних звуків, таких як розбите скло, тривожні вигуки, або несподівана активність у неочікуваний час.

Кожен з цих компонентів підключається до Arduino UNO через відповідні вхідні/вихідні порти і може вимагати додаткових елементів, таких як опори, транзистори або релейні модулі для керування навантаженнями. Також важливо врахувати необхідність написання відповідного програмного коду для зчитування даних з датчиків та їх обробки.

Встановлення та налаштування модулів системи «Розумний дім» починається з підключення та конфігурації набору датчиків до мікроконтролера Arduino UNO. Кожен датчик - PIR D203S для виявлення руху, DHT22 для моніторингу температури та вологості, Reed module KY-025 для визначення стану вікон та дверей, фоторезистор KY-018 для вимірювання інтенсивності світла та мікрофонний модуль MAX9814 для аудіомоніторингу - є критичними у виявленні фізичних змін у домовому середовищі.

Кожен з цих датчиків відповідає за збір конкретного типу даних. Наприклад, датчик PIR D203S активізується при виявленні руху, що може сигналізувати про присутність людей або тварин в зоні детекції. DHT22 вимірює мікроклімат у приміщенні, генеруючи дані, які можуть бути використані для автоматичного регулювання опалення чи кондиціонування. Reed module KY-025 є ключовим у системах безпеки, відстежуючи відкриття та закриття доступів до будинку. Фоторезистор KY-018 дозволяє системі адаптуватися до змін у освітленні, тоді як мікрофонний модуль MAX9814 надає аудіоаналітику для виявлення звукових аномалій.

Arduino UNO, використовуючи власний програмний код, зчитує сигнали з датчиків, обробляє їх та перетворює у зрозумілі для системи формати даних. Ці дані потім відправляються на сервер, де вони зберігаються у базі даних. Сервер обробляє вхідні дані, застосовуючи алгоритми на основі LSTM для виявлення потенційних загроз в реальному часі. Це може включати ідентифікацію незвичайної активності, такої як несподіваний рух у певний час доби або аномалії у температурних показниках, які можуть вказувати на ризик пожежі чи інші небезпечні ситуації.

Додаток, розроблений для взаємодії з цією базою даних, надає користувачам інтерфейс для моніторингу стану їхнього дому, відображаючи актуальну інформацію та сповіщення про будь-які виявлені загрози. Такий підхід не тільки забезпечує поточну безпеку, але й збирає цінні дані, які можуть бути аналізовані для подальшого удосконалення моделей LSTM, дозволяючи системі адаптуватися та вдосконалюватися з часом, виявляючи нові шаблони поведінки та реагуючи на еволюцію зовнішніх умов.

На рис. 2.11 показано загальну схему взаємодії всіх компонентів системи.

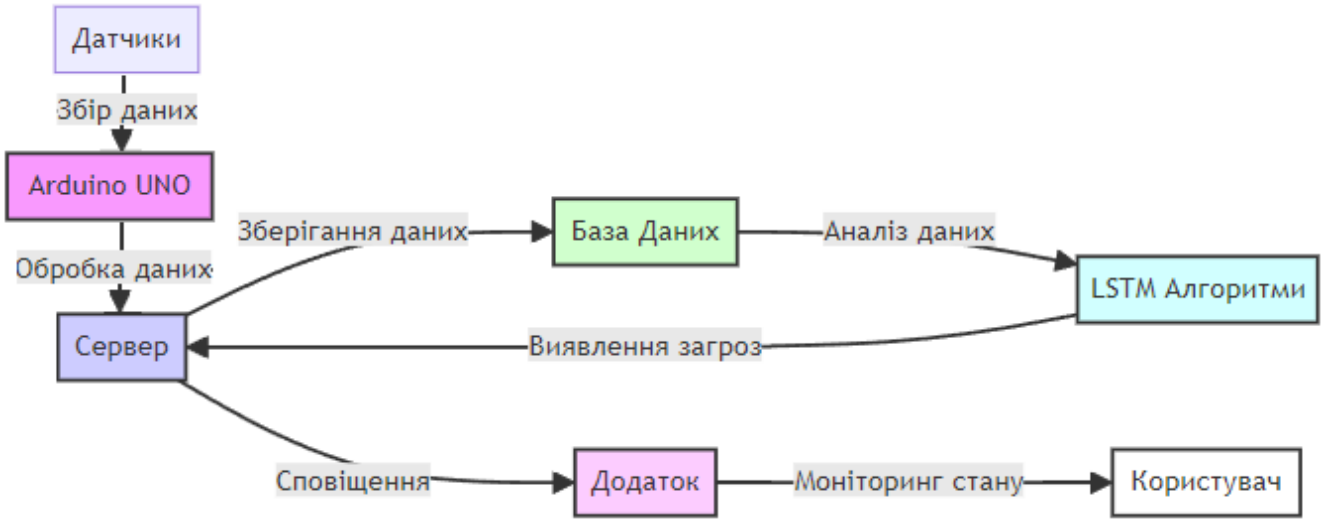


Рисунок 2.11– Схема взаємодії компонентів системи

Таким чином, кожен модуль і датчик не лише виконує свою ізольовану функцію, але й стає частиною більшої інтегрованої системи, де дані та аналітика об'єднуються для створення безпечного, інтелектуального та реактивного домашнього середовища.

2.5 Висновок до розділу

У рамках розділу було зосереджено увагу на проектуванні інтелектуальної системи безпеки, що ґрунтується на використанні LSTM мереж. Під час розробки інтелектуальної системи будинку була створена математична модель, яка дозволяє прогнозувати поведінкові шаблони користувачів, що є важливим для превентивних заходів безпеки. При виборі та проектуванні архітектури системи було вирішено зупинитися на трьохрівневій структурі, що забезпечує необхідну гнучкість та масштабованість.

Для ідентифікації та класифікації потенційних загроз була розроблена блок-схема алгоритму, що є фундаментальною для розуміння логіки роботи системи та її реакції на змінні умови. Вибір комплектуючих для системи, включаючи контролер Arduino UNO та набір датчиків, був здійснений на основі чітко визначених критеріїв, що забезпечує оптимальне співвідношення функціональності, ціни та якості.

Розділ закладає основу для практичної реалізації системи «Розумний дім», визначаючи ключові напрямки для подальшої розробки та інтеграції. Результати проектування підкреслюють важливість комплексного підходу до забезпечення безпеки розумного будинку, де кожен компонент відіграє важливу роль в створенні надійної, зв'язної та інтелектуальної системи, готової відповідати на сучасні виклики.

3 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ АЛГОРИТМУ ІДЕНТИФІКАЦІЇ ТА НЕЙТРАЛІЗАЦІЇ КІБЕРЗАГРОЗ У СИСТЕМІ «РОЗУМНИЙ ДІМ»

3.1 Налаштування експериментального середовища

Початок роботи з Arduino UNO для розробки системи «розумний дім» розпочинається з встановлення Arduino Integrated Development Environment (IDE), яке є ключовим інструментом для програмування мікроконтролерів від Arduino. Після завантаження та інсталяції Arduino IDE, наступним кроком є підключення Arduino UNO до комп'ютера через USB-кабель. Це підключення активує плату, про що свідчить загоряння світлодіоду «ON» та блимання світлодіоду «L», що вказує на виконання програми Blink.

Для подальшої роботи з платою необхідно з'ясувати, який номер порту COM комп'ютер привласнив Arduino UNO. Це можна виявити, відкривши Диспетчер пристроїв Windows та переглянувши вкладку «Порти (COM і LPT)», де вказаний відповідний порт для Arduino UNO (рис. 3.1). Цей номер порту COM використовується при налаштуванні Arduino IDE для встановлення зв'язку з платою.

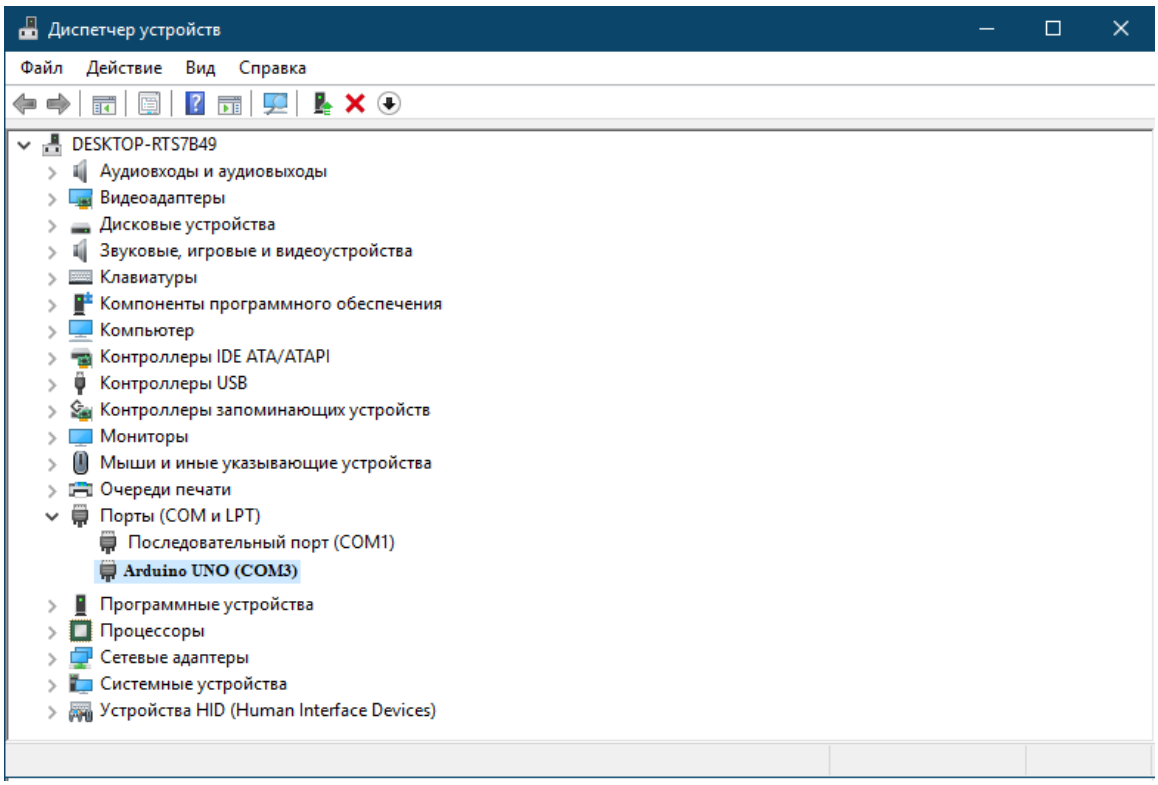


Рисунок 3.1 – Диспетчер пристроїв Windows для виявлення порту Arduino UNO

Коли плата Arduino UNO вперше підключається до комп'ютера, операційна система виконує процес виявлення та ідентифікації нового пристрою. У даному випадку, система успішно розпізнала Arduino UNO як COM-порт, автоматично підібрала та встановила необхідний драйвер, і присвоїла цьому порту номер, наприклад, «COM3». Важливо зазначити, що кожен раз при підключенні нової плати Arduino до комп'ютера, операційна система може призначити їй унікальний номер COM-порту. Таким чином, якщо одночасно використовуються декілька плат Arduino, кожна з них матиме свій індивідуальний номер порту. Наступним кроком у процесі налаштування є інформування Arduino IDE про те, що плата Arduino UNO, з якою потрібно взаємодіяти, знаходиться на порту «COM3». Для цього, у самій програмі Arduino IDE користувачу необхідно відкрити меню «Сервіс», вибрати розділ «Послідовний порт» і там встановити порт «COM3» (рис. 3.2). Після цього, Arduino IDE буде точно знати, що контролер підключений до порту «COM3». Це дає можливість IDE правильно налаштовувати середовище для завантаження програм на контролер і, водночас, забезпечувати моніторинг цього порту для отримання даних від контролера.

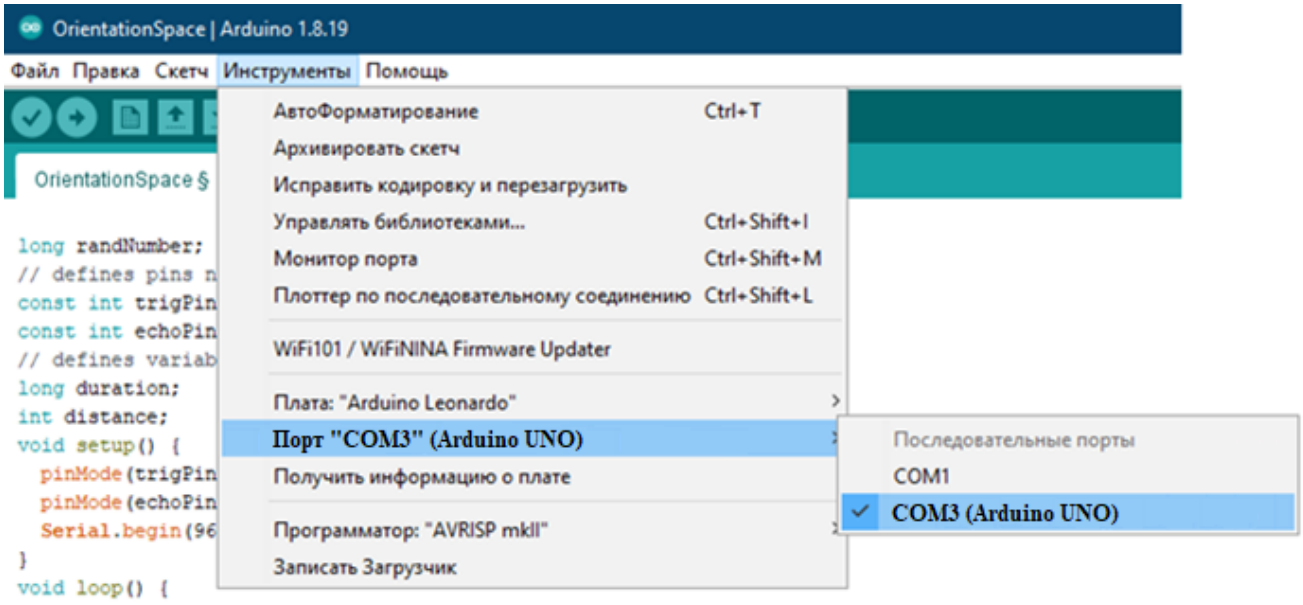


Рисунок 3.2 – Вибір порту з’єднання з Arduino UNO

Цей процес є важливим для ефективної взаємодії між програмним забезпеченням та апаратною частиною, оскільки забезпечує синхронізацію між

кодом, написаним користувачем, і фізичною роботою плати Arduino в реальному часі.

Після успішного визначення та налаштування номера COM-порту для плати Arduino UNO, важливим наступним кроком у процесі роботи з Arduino IDE є вказівка конкретного типу контролера, який буде використовуватись. Цей процес забезпечує відповідність між середовищем програмування та апаратними характеристиками плати, що є вирішальним для правильної компіляції та завантаження програми. Щоб вказати, що використовується саме Arduino UNO, необхідно відкрити меню «Сервіс» у програмі Arduino IDE і перейти у розділ «Плата», де серед переліку доступних плат слід вибрати «Arduino UNO». Це вказівка дозволяє Arduino IDE коректно налаштувати всі необхідні параметри для роботи з цією конкретною моделлю плати.

Після цих налаштувань настає етап написання та тестування програми. Програмування в Arduino IDE здійснюється з використанням вбудованої мови, яка базується на C/C++. Коли програма написана, вона має бути скомпільована, щоб перевірити наявність помилок та готовність до завантаження на плату. Для компіляції програми в Arduino IDE використовується комбінація клавіш «Ctrl+R». Під час цього процесу, програма перевіряється на синтаксичні та логічні помилки. Якщо в процесі компіляції помилок не виявлено, внизу вікна Arduino IDE з'являється повідомлення «Компіляція завершена». Це повідомлення свідчить про те, що програма готова до завантаження на плату та подальшого виконання.

Завершення успішної компіляції є ключовим моментом у процесі розробки, оскільки воно підтверджує, що написана програма не містить помилок, що можуть перешкодити її виконанню на фізичній платі Arduino. Успішна компіляція також забезпечує впевненість у тому, що всі команди та функції, використані у програмі, є підтримуваними та коректно виконуються в межах апаратних можливостей плати Arduino UNO (рис. 3.3).

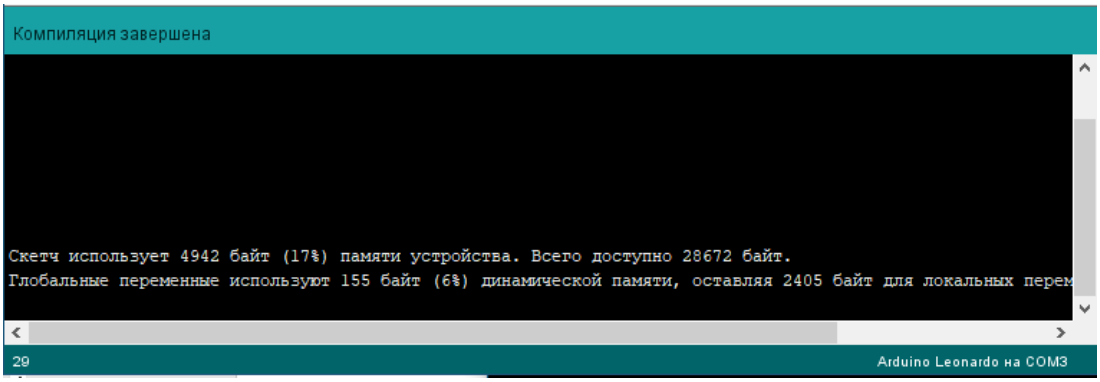


Рисунок 3.3 – Компіляція розробленого скетчу на стороні контролера

Завершивши процес компіляції проекту без помилок, настає час завантажити програму безпосередньо на контролер Arduino. Це здійснюється через інтерфейс Arduino IDE, де потрібно виконати кілька кроків. Спершу, в меню програми необхідно знайти вкладку «Скетч». Під цим пунктом знаходиться опція «Завантаження», яка відповідає за процес передачі скомпільованої програми з комп'ютера на мікроконтролер Arduino. Вибравши цю опцію, розпочнеться процес завантаження програми на плату, що є фінальним етапом перед її виконанням в реальному часі (рис. 3.4).

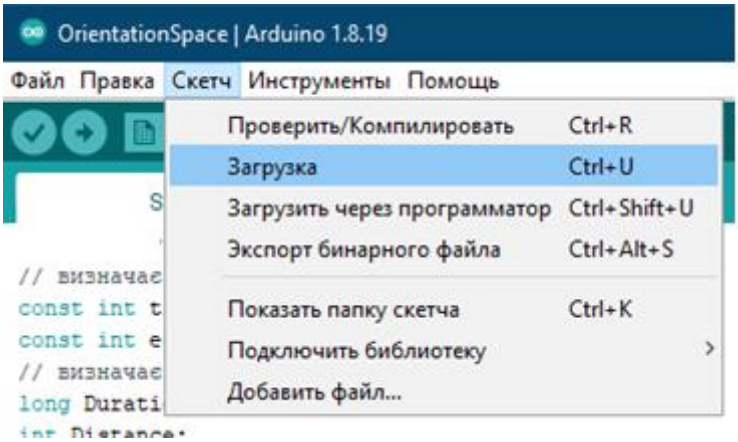


Рисунок 3.4 – Завантаження програми

В момент, коли процес передачі програми з комп'ютера на контролер Arduino завершується, користувач отримує візуальне підтвердження цього успішного завантаження. В інтерфейсі Arduino IDE в нижній частині вікна з'являється інформаційне повідомлення, яке підтверджує, що програма була коректно завантажена на контролер. Це повідомлення є важливим індикатором успішності процесу, оскільки воно засвідчує, що програма не лише була правильно скомпільована, але й успішно інтегрована в апаратну частину системи. Для

зручності користувачів, візуальне відображення цього повідомлення можна побачити на ілюстрації, представленій на рис. 3.5.

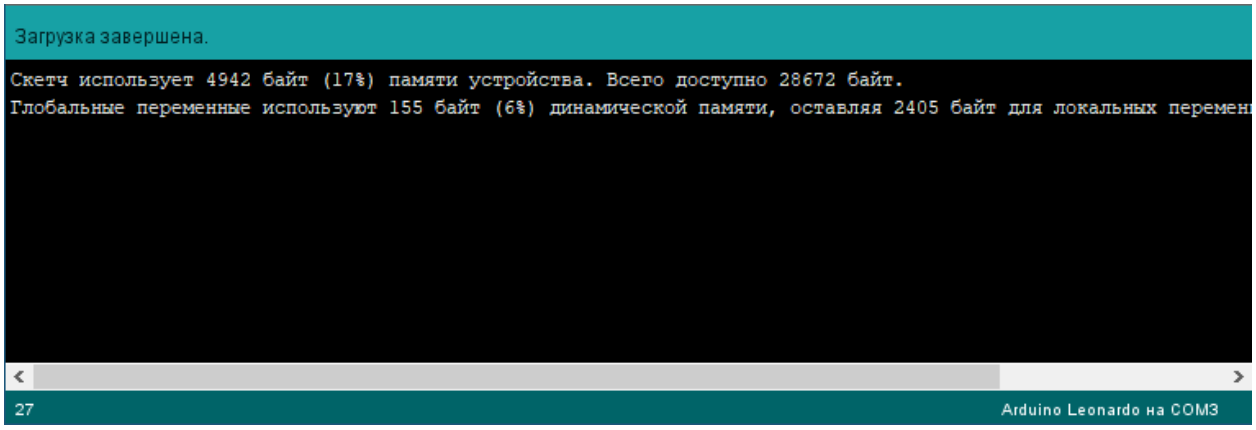


Рисунок 3.5 – Повідомлення про завантаження програми на контролер

3.2 Збір даних та проведення навчання моделі

Для реалізації дослідження було вибрано специфічний набір даних із платформи Kaggle, яка є відомим ресурсом для науковців та розробників у галузі машинного навчання. Набір даних, доступний за посиланням [41], надає унікальну можливість аналізувати поведінкові шаблони в контексті розумного будинку. Він містить об'ємні дані, зібрані з множини датчиків, встановлених у розумному будинку, включаючи, але не обмежуючись, сенсорами руху, температури, вологості, звукового тиску, а також датчиками, що фіксують зміни в освітленні та світловому середовищі. Ці дані відіграють критичну роль у розумінні поведінкових патернів та забезпеченні безпеки в системах розумного будинку.

Оскільки мережі LSTM вимагають високої якості та консистентності даних для ефективного навчання та прогнозування, підготовка даних включала ретельну нормалізацію та прекондиціонування. Спочатку дані були проаналізовані на предмет відхилень та аномалій. Нормалізація полягала у вирівнюванні різних масштабів даних, що забезпечує однорідність та збереження корисної інформації для навчання мережі. Далі, процес фільтрації даних включав видалення шумів та ірелевантних даних, що могли негативно вплинути на точність моделі. Зокрема, було відфільтровано випадкові шуми та виправлено можливі помилки в даних, пов'язані з нештатними умовами роботи датчиків. Завершальний крок обробки

включав стандартизацію форматів даних та їхню підготовку до введення в мережу LSTM. Цей комплексний підхід до підготовки даних забезпечив оптимальне середовище для навчання мережі та подальшого точного прогнозування.

На рис. 3.6 представлено фрагмент вікна із підготовленими даними для проведення навчання моделі.

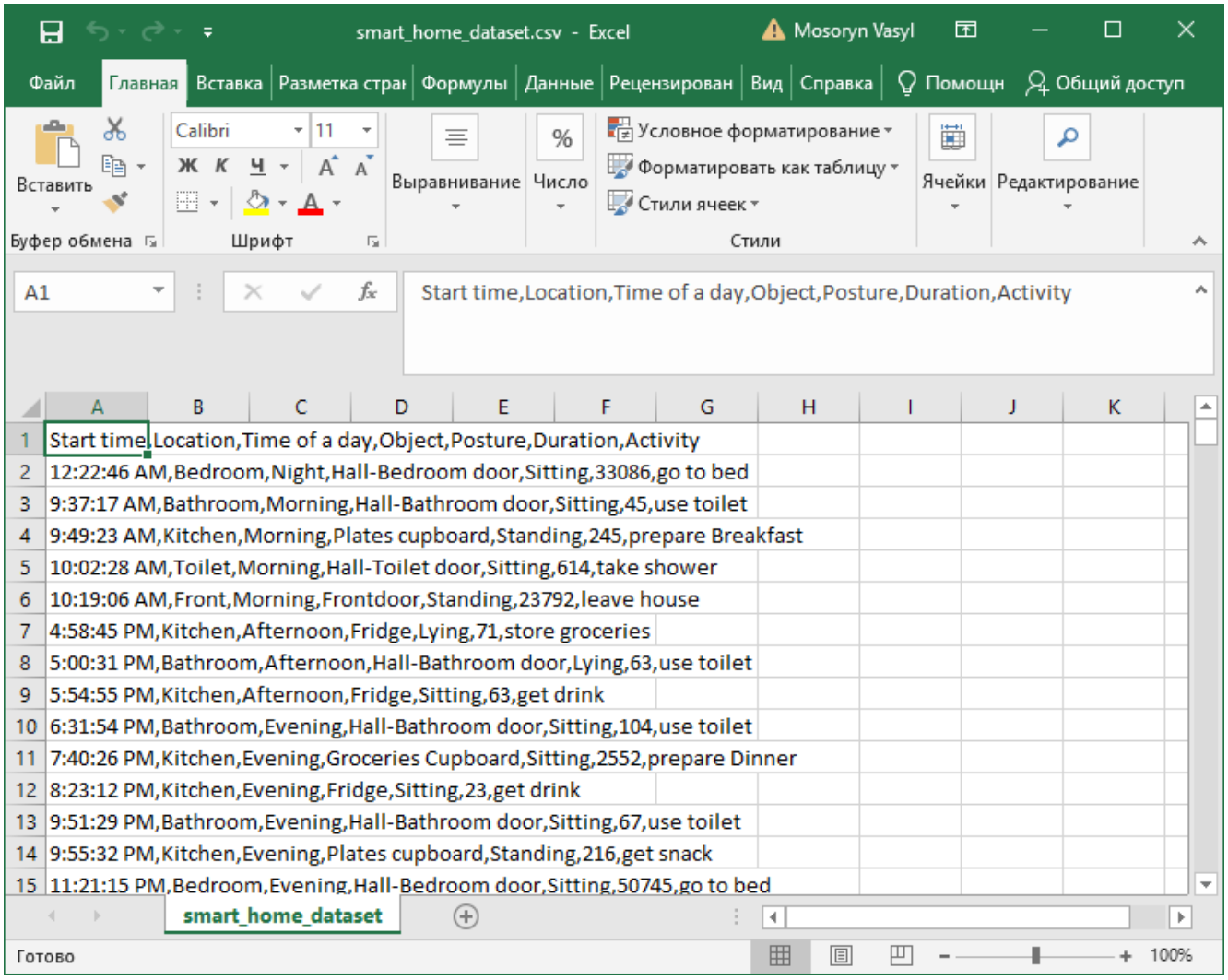


Рисунок 3.6 – Фрагмент вікна із підготовленими даними для навчання моделі

Для тренування моделі виявлення вторгнень за допомого мережі LSTM у застосунку було створено категорію для навчання із назвою «Категорія загроз №1» (рис. 3.7).

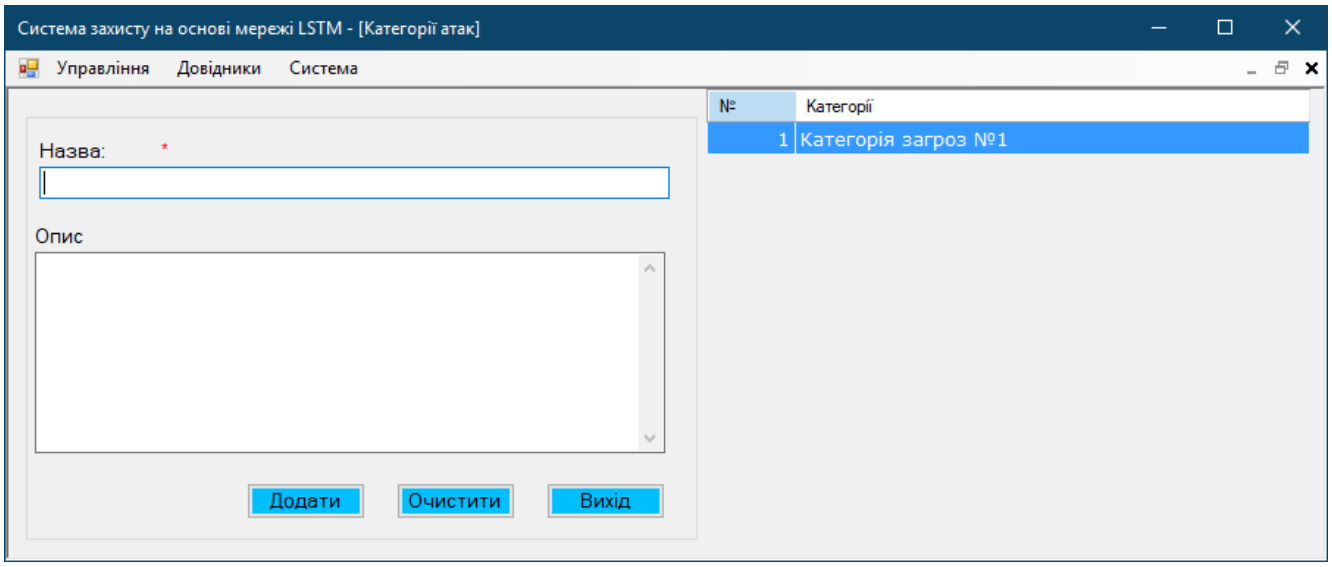


Рисунок 3.7– Створення категорії навчання моделі

Для тренування нової моделі для створеної категорії згідно підготовленого датасету, було здійснено перехід по меню програми «Довідники» –> «Тренування моделей» та обрано створену категорію «Категорія загроз №1». Після обрання підготовленого датасету у діалоговому вікні навчання нейронної мережі почалось автоматично. Результат навчання представлено на рис. 3.8.

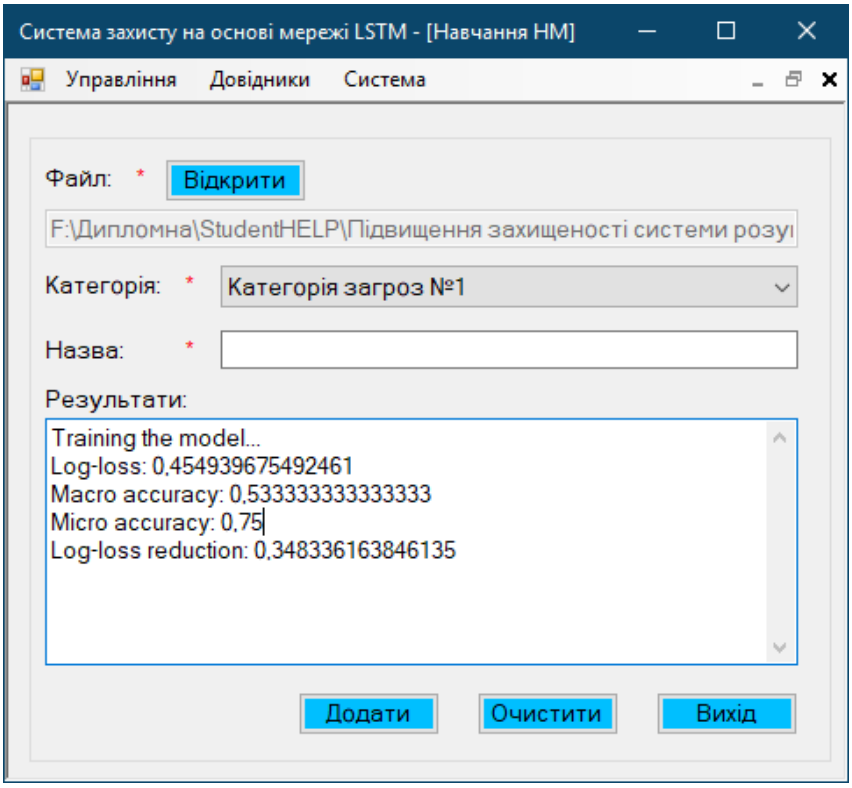


Рисунок 3.8 – Результат навчання НМ

Процес навчання НМ супроводжується виведенням повідомлення «Training the model...» та виведенням значень метрик навчання:

– Log-loss (Логарифмічна втрата). Значення log-loss становить 0,454939675492461. Це показник, який використовується для оцінки точності прогнозів моделі, де нижчі значення вказують на кращу якість. В даному випадку, log-loss менше 0,5 свідчить про те, що прогнози моделі відносно точно відповідають фактичним етикеткам. Це досить хороший показник, що вказує на високу точність моделі у прогнозуванні;

– Macro Accuracy (Макро-точність). Значення макро-точності не змінилося і становить 0,533333333333333. Це середнє значення точності по всіх класах, яке вказує на середній рівень точності моделі у виявленні різних класів. Модель може демонструвати нерівномірну продуктивність по різних класах, що може бути вказівкою на необхідність подальшого налаштування або збалансування даних;

– Micro Accuracy (Мікро-точність). Значення мікро-точності також залишається незмінним – 0,75. Цей показник вимірює загальну точність класифікації, беручи до уваги всі класи. Точність у 75% є досить високою, вказуючи на ефективність моделі у правильному класифікуванні випадків;

– Log-loss Reduction (Зниження логарифмічної втрати): Значення log-loss reduction залишається незмінним і становить 0,348336163846135. Це показник покращення моделі порівняно з базовим рівнем log-loss і вказує на те, що модель виявилася ефективнішою порівняно з базовим сценарієм.

З огляду на ці показники, модель демонструє досить хорошу ефективність з точки зору загальної точності (micro accuracy) і відносно низької логарифмічної втрати (log-loss). Хоча макро-точність залишається на середньому рівні, загальна ефективність моделі та її спроможність знизити логарифмічну втрату свідчать про високу якість прогнозування.

Після цього навчену модель було збережено у системі, для проведення експериментального дослідження (рис. 3.9).

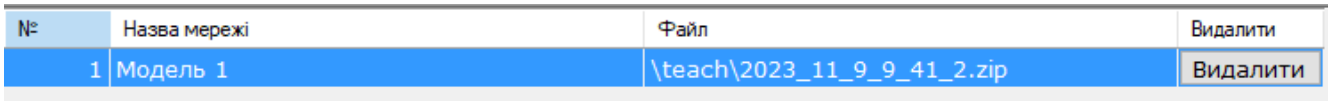


Рисунок 3.9 –Збереження моделі НМ у системі

3.3 Сценарії потенційних атак та їх симуляція

Для проведення тестування збереженої моделі необхідно здійснити перехід по меню програми «Управління» – > «Тестування моделей». У формі, що відкриється можна проводити моделювання потенційних сценаріїв вторгнення в систему «розумний дім». Користуючись випадаючим списком можна обрати необхідну категорію для якої було проведено навчання моделі за допомогою мережі LSTM.

Процес прогнозування потенційних загроз для системи «Розумний дім» відбувається у такі етапи:

– генерація Даних. Програма включає клас розроблений клас (SmartHomeDataGenerator), який відповідає за генерацію випадкових даних, які імітують повсякденні події в домі та можливі вторгнення. Для цього використовується набір заздалегідь визначених параметрів, таких як локації (наприклад, спальня, ванна кімната), час доби, об'єкти (наприклад, двері, вікна), пози (сидячи, стоячи) та види діяльності (наприклад, «йти спати», «використовувати туалет»). Зокрема, включені спеціалізовані діяльності для симуляції вторгнення, такі як «віджати двері», «розбити вікно», «несанкціонований вхід».

– прогнозування за допомогою моделі. Коли дані згенеровані, вони передаються в нейронну мережу для аналізу. Мережа оцінює кожну подію та визначає ймовірність того, що конкретна подія може бути вторгненням;

– аналіз результатів прогнозування. Результати, отримані від нейронної мережі, аналізуються програмою для визначення, чи відбувається потенційне вторгнення. Це включає оцінку передбачуваної діяльності та визначення, чи є ця діяльність типовою для даного контексту чи може свідчити про загрозу;

– візуалізація та звітування: Результати симуляції відображаються в спеціальному текстовому полі програми. Кожен запис містить інформацію про

згенеровані дані, включаючи час, місцезнаходження, час доби, об'єкт, позу, тривалість, діяльність, а також передбачувану діяльність та оцінку потенційного вторгнення.

Основна мета цієї симуляції полягає в тому, щоб перевірити здатність розробленої нейронної мережі ефективно розрізняти звичайні події домашнього життя від потенційних загроз. Це не тільки демонструє ефективність розробленої системи, але й дає змогу виявити та усунути можливі недоліки в моделі перед її реальним застосуванням.

Для виявлення точності навченої моделі за допомогою мережі LSTM було згенеровано декілька потенційних сценаріїв за допомогою розробленого генератора сценаріїв та детально їх проаналізовано. На рис. 3.10 зображено перший згенерований сценарій.

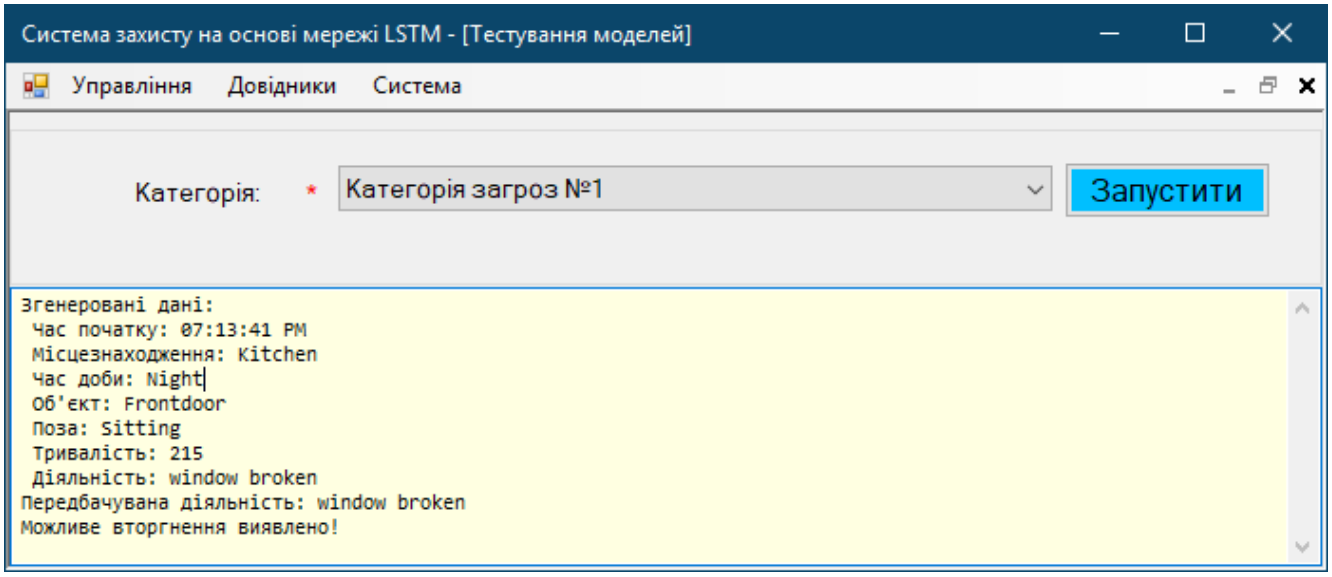


Рисунок 3.10 –Результат генерації 1-го сценарію

Аналізуючи наданий сценарій, можна виділити кілька ключових аспектів, які свідчать про потенційне вторгнення:

– час та місцезнаходження. Подія відбувається о 07:13:41 вечора у кухні. Хоча це і не нічний час, проте вечірня пора часто асоціюється зі зниженою активністю в домі, що може створювати сприятливі умови для непоміченого вторгнення;

– об'єкт та поза. Об'єкт діяльності – «Frontdoor» (фронтальні двері), а поза – «Sitting» (сидяча). Це може свідчити про те, що особа перебувала неподалік

входу, що є нетиповим у контексті кухні та може вказувати на незвичайну поведінку або ситуацію;

– тривалість. Діяльність тривала 215 секунд (близько 3,5 хвилин), що є досить тривалим періодом для ситуації, яка потенційно може включати вторгнення;

– діяльність та передбачувана діяльність. Зазначено, що відбувається діяльність «window broken» (розбите вікно), яка також визначена як передбачувана діяльність. Це вказує на збіг фактичних та модельно прогнозованих подій, що може бути чітким індикатором вторгнення;

– оцінка ситуації. Подальший аналіз системою визначає, що «Можливе вторгнення виявлено!». Це означає, що на основі аналізу зібраних даних, система змогла коректно ідентифікувати потенційну загрозу вторгнення.

Отже, сценарій №1 демонструє ефективність системи у виявленні реальних загроз. Явне виявлення розбитого вікна та визначення цієї події як потенційного вторгнення свідчать про високу точність моделі у виявленні та правильному реагуванні на загрози. Це свідчить про успішне навчання та калібрування моделі, здатної точно оцінювати різні ситуації в домашньому середовищі та ефективно реагувати на них.

На рис. 3.11 зображено скріншот результат виконання 2-го сценарію.

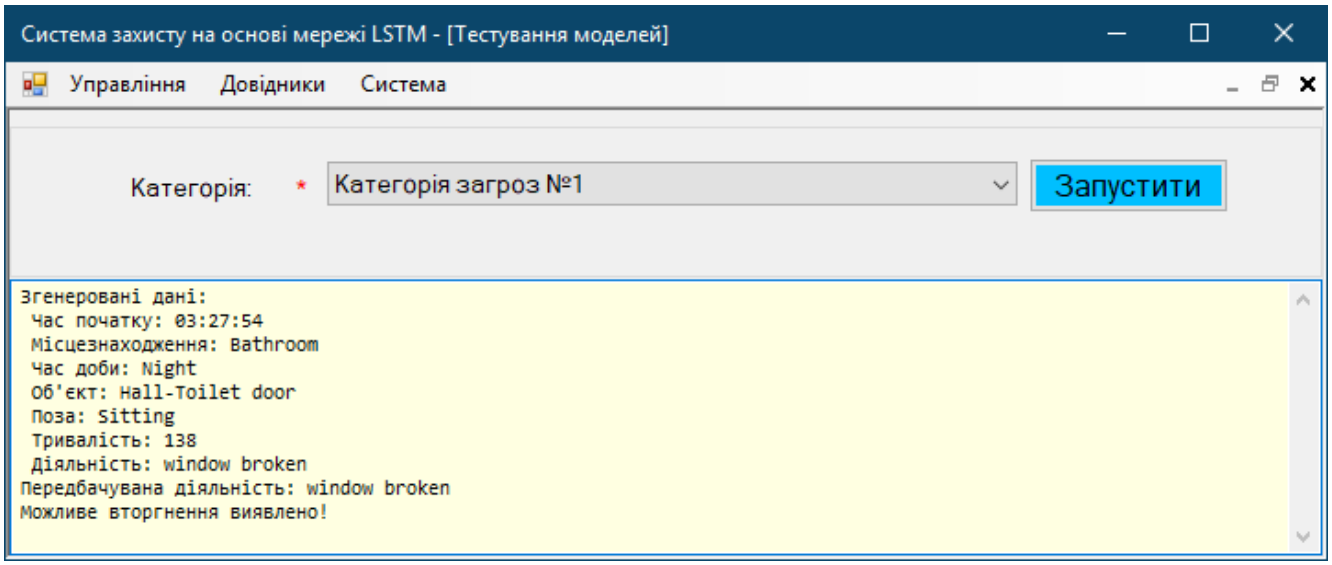


Рисунок 3.11 –Результат генерації 2-го сценарію

Аналізуючи наданий сценарій із згенерованими даними, можна зробити наступні висновки:

– час та місцезнаходження. Подія відбувається о 03:27:54, що вказує на ранній нічний час. Локація – ванна кімната. Нічний час є критичним фактором, оскільки більшість вторгнень відбувається вночі, коли є менше шансів бути поміченими;

– об'єкт та поза. Об'єкт діяльності – двері між холом та туалетом, з пози «Sitting» (сидяча). Це може вказувати на те, що особа знаходиться в стані очікування або відпочинку, але також може бути нетиповим для подібної ситуації, особливо в контексті «window broken»;

– тривалість. Діяльність триває 138 секунд (близько 2 хвилин і 18 секунд), що є відносно коротким періодом часу для будь-якої активності, але достатнім для вчинення вандалізму або вторгнення;

– діяльність та передбачувана діяльність. Вказана діяльність «window broken» (розбите вікно), яка також визначена як передбачувана діяльність. Це вказує на то, що система точно виявила потенційно небезпечну ситуацію та правильно ідентифікувала її характер;

– оцінка ситуації. Виявлення «Можливого вторгнення» свідчить про ефективність системи у виявленні потенційних загроз на основі згенерованих даних. Це підтверджує, що модель здатна коректно реагувати на виявлені аномалії у поведінці або навколишньому середовищі.

Отже, цей сценарій демонструє, що система ефективно використовує згенеровані дані для виявлення потенційних загроз, таких як вторгнення, на основі комбінації часу, місцезнаходження, об'єкта, пози, тривалості та виду діяльності. Точність системи у визначенні «window broken» як потенційного вторгнення підтверджує її надійність та ефективність у контексті захисту систем «розумного дому».

Рис. 3.12 відображає скріншот результату виконання 3-го сценарію проведеного експерименту.

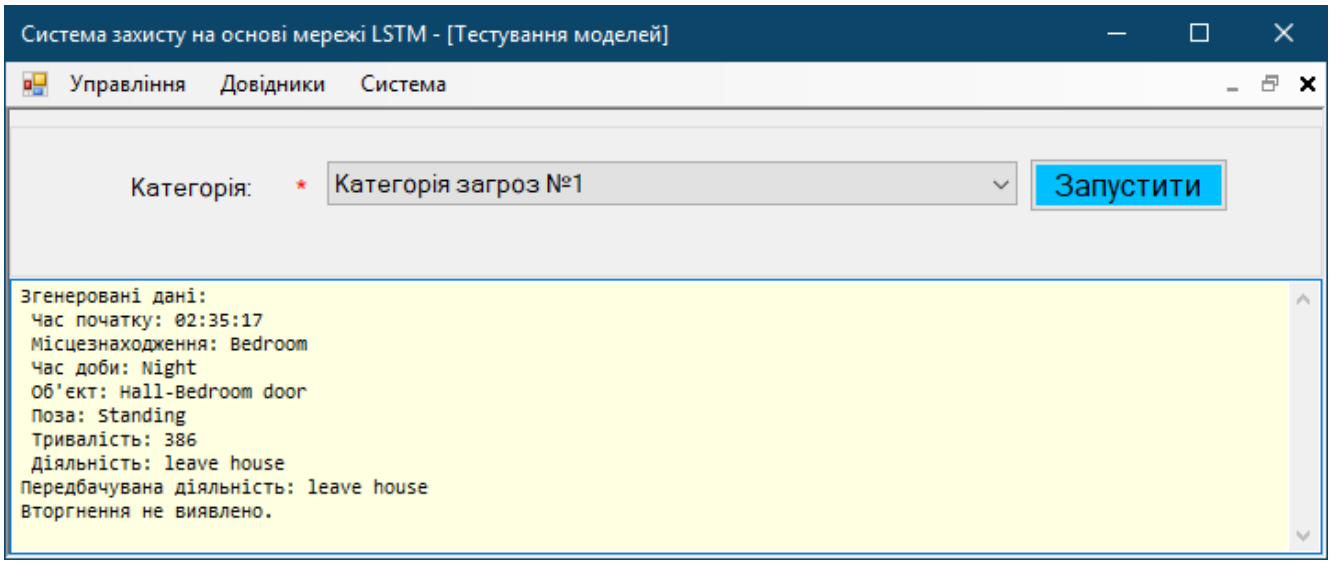


Рисунок 3.12 –Результат генерації 3-го сценарію

Аналізуючи наданий сценарій згенерованих даних, можна виділити наступні ключові аспекти:

– час та місцезнаходження. Згенеровані дані вказують на час початку о 02:35:17, що відповідає раннім нічним годинам. Місцезнаходження – спальня. Це час, коли зазвичай очікується спокій у домашніх умовах, а спальня є місцем для відпочинку;

– об'єкт та поза. Об'єкт діяльності – двері між холом та спальнею, з пози «Standing» (стояча). Це може вказувати на активність або підготовку до якоїсь дії біля дверей спальні;

– тривалість. Діяльність триває 386 секунд (близько 6 хвилин і 26 секунд), що є значним часом для будь-якої діяльності, особливо вночі;

– діяльність та передбачувана діяльність. Фактична та передбачувана діяльність обидві вказують на «leave house» (залишення будинку). Це вказує на те, що система правильно визначила намір особи покинути будинок;

– оцінка ситуації. Незважаючи на те, що діяльність «залишення будинку» відбувається в нічний час, система визначила, що «Вторгнення не виявлено». Це може вказувати на те, що система розглядає цю діяльність як звичайну або не підозрілу, можливо, через відсутність інших ознак вторгнення або незвичайної активності.

Отже, 3-й сценарій демонструє, що система здатна відрізняти нормальні домашні діяльності від потенційних вторгнень, навіть коли діяльність відбувається у нетиповий для неї час. Відсутність ідентифікації вторгнення в цьому випадку може вказувати на те, що система враховує конкретний контекст діяльності, такий як час доби та місцезнаходження, перш ніж робити висновок про потенційну загрозу. Однак, це також може вказувати на потребу в додатковому аналізі або вдосконаленні моделі для точнішого виявлення випадків вторгнення в нетипові часи.

3.4 Аналіз отриманих результатів та їх валідація

У рамках проведення тестування збереженої моделі за допомогою мережі LSTM було згенеровано декілька потенційних сценаріїв вторгнення в систему «розумний дім». Це дозволило провести детальний аналіз ефективності моделі у виявленні реальних загроз.

Процес аналізу включав кілька ключових етапів:

– генерація даних. За допомогою класу SmartHomeDataGenerator, були створені сценарії, імітуючі різноманітні події в домашньому середовищі, включаючи потенційні вторгнення;

– прогнозування за допомогою моделі: Генеровані дані були проаналізовані за допомогою нейронної мережі, яка оцінювала ймовірність кожної події як потенційного вторгнення;

– аналіз результатів прогнозування. Результати, отримані від мережі, були проаналізовані для визначення потенційних вторгнень, враховуючи передбачувану діяльність та контекст подій;

– візуалізація та звітування. Результати були відображені в текстовому полі програми для зручності візуального аналізу.

Щодо отриманих результатів у процесі проведеного експериментального дослідження, можна зробити такі висновки:

– перший сценарій. Зафіксовано вечірній час у кухні, з незвичайною активністю біля фронтальних дверей, що вказувало на потенційне вторгнення.

Сценарій показав високу точність моделі у виявленні загроз, зокрема, розбитого вікна як індикатора вторгнення;

– другий сценарій. Відбувається раннього нічного часу у ванній кімнаті, з сидячою позою біля дверей, що також вказувало на потенційне вторгнення. Модель точно ідентифікувала «розбите вікно» як загрозу;

– третій сценарій. Нічний час у спальні з активністю біля дверей, що вказувало на підготовку до залишення будинку. Модель коректно визначила відсутність вторгнення, що свідчило про її здатність розрізняти звичайні домашні діяльності від потенційних загроз.

Щодо проведеного аналізу розробленої системи та експериментальної її перевірки можна виділити наступні особливості системи:

– ефективність прогнозування. Процес прогнозування потенційних загроз виявився ефективним. Модель змогла адекватно інтерпретувати згенеровані дані, виявляючи потенційні вторгнення на основі комбінації різних параметрів, таких як місцезнаходження, час доби, об'єкт, поза, тривалість, та вид діяльності. Це підтверджує здатність моделі розрізняти звичайні події від потенційних загроз;

– точність ідентифікації. У кожному із сценаріїв, модель показала високу точність у визначенні природи діяльності, правильно ідентифікуючи як звичайні домашні діяльності, так і потенційні вторгнення. Наприклад, в одному зі сценаріїв модель вірно ідентифікувала «розбите вікно» як індикатор потенційного вторгнення;

– контекстуальний аналіз. Аналіз показав, що система здатна враховувати контекстуальні особливості, такі як час доби та місцезнаходження, перед тим як робити висновок про потенційну загрозу. Це свідчить про розвинені можливості системи в аналізі комплексних ситуацій;

– виявлення нетипової діяльності. Система ефективно виявляла нетипові діяльності, такі як «залишення будинку» вночі, без помилкового ідентифікування цих дій як вторгнення. Це демонструє важливість балансу між чутливістю та специфічністю в моделях прогнозування;

– потенціал для покращення. Попри загалом високу точність, існують моменти для подальшого вдосконалення, особливо у випадках, коли діяльність відбувається в нетиповий час. Моделі потрібно бути здатними точніше визначати ризик на основі ширшого спектру даних.

Отже, проведення валідації результатів показало, що розроблена система є надійною та ефективною, її можна використовувати для виявлення потенційних загроз.

3.5 Шляхи подальшого вдосконалення та розвитку системи

Розроблена система захисту на основі мережі LSTM, відкриває нові горизонти для захисту та комфорту в повсякденному житті. Проте, для реалізації повного потенціалу цієї системи, важливо зосередитись на її подальшому вдосконаленні та розвитку. З цією метою, ідентифіковано ключові напрямки, які можуть значно підсилити її ефективність та гнучкість:

– інтеграція з додатковими датчиками та джерелами даних. Залучення більшої кількості датчиків та інших джерел даних може покращити здатність системи до виявлення аномалій та прогнозування потенційних кіберзагроз. Наприклад, інтеграція з системами відеоспостереження, сенсорами звуку, чи датчиками якості повітря може дати LSTM-мережам більше контексту для аналізу;

– розробка гібридних моделей ШІ. Комбінування LSTM з іншими моделями штучного інтелекту, такими як згорткові нейронні мережі (CNN) для обробки візуальних даних, може збільшити ефективність системи. Гібридні моделі можуть краще обробляти різноманітні види даних та забезпечувати більш точний аналіз;

– адаптивне навчання та самовдосконалення. Розробка механізмів для постійного навчання та адаптації моделі LSTM на основі нових даних та змін у поведінці користувачів. Це дозволить системі бути більш гнучкою та ефективною у виявленні нових видів загроз;

– забезпечення приватності та безпеки даних. Оскільки системи розумного будинку збирають великі обсяги особистих даних, важливо

вдосконалити механізми захисту цих даних. Впровадження сучасних методів шифрування та анонімізації даних може підвищити рівень довіри та безпеки системи;

– інтеграція з екосистемами розумного будинку. Інтеграція системи з іншими компонентами розумного будинку, такими як системи управління освітленням, опаленням, та іншими побутовими пристроями, може дозволити створити більш інтегровану та автоматизовану систему безпеки;

– використання хмарних обчислень та великих даних. Використання хмарних платформ для зберігання, обробки та аналізу великих обсягів даних може підвищити швидкість та ефективність системи. Це також може дозволити впровадити більш складні алгоритми ШІ, які вимагають значних обчислювальних ресурсів;

– формування співтовариств та обмін даними. Створення мережі між системами розумних будинків для обміну даними про виявлені загрози та патерни поведінки може допомогти в швидшому виявленні та нейтралізації загроз;

– стандартизація та сертифікація. Розробка стандартів та процедур сертифікації для систем розумного дому на основі LSTM може сприяти їх ширшому прийняттю та використанню на ринку;

– подальші дослідження в галузі кібербезпеки. Продовження досліджень у сфері кібербезпеки, зокрема в аспектах застосування штучного інтелекту, може допомогти виявити нові можливості та методи захисту від кіберзагроз.

3.6 Висновок до розділу

У рамках даного розділу було проведено грунтовне експериментальне дослідження та оцінка ефективності пропонованої системи безпеки розумного дому на основі LSTM мереж. Значною частиною роботи стало налаштування експериментального середовища, збір та підготовка даних, що включали ретельну нормалізацію та прекондиціонування для забезпечення високої якості та консистентності необхідної для ефективного навчання LSTM мережі. Отримані

метрики моделі, такі як log-loss, макро- та мікро-точність, свідчили про високу ефективність системи у виявленні потенційних кіберзагроз.

Симуляція сценаріїв потенційних атак підтвердила, що розроблена система спроможна точно ідентифікувати різні види загроз. Перший сценарій продемонстрував високу точність виявлення вторгнення через розбите вікно, другий сценарій підтвердив ефективність системи у виявленні потенційного вторгнення у ванній кімнаті, а третій показав здатність системи розрізняти звичайні домашні діяльності від загроз. Ці результати підтверджують високий рівень надійності та адаптивності системи.

Аналіз отриманих результатів та їх валідація підкреслюють наукову та практичну цінність дослідження. Воно не лише відкриває шляхи для подальших досліджень у сфері кібербезпеки й розвитку інтелектуальних систем, але й становить значний вклад у розвиток концепцій інтелектуальної безпеки розумних будинків. Використання отриманих результатів у практиці дозволить створювати більш ефективні та адаптивні системи безпеки, здатні протистояти широкому спектру кіберзагроз.

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Магістерська кваліфікаційна робота є науково-технічною роботою, що орієнтована на комерціалізацію. Вона включає розробку інноваційної комплексної системи «Розумний дім», заснованої на застосуванні мережі штучного інтелекту LSTM для виявлення вторгнень. Основна мета проекту полягає у створенні високотехнологічного продукту, який може бути виведений на ринок та використаний у комерційних цілях.

Система «Розумний дім» поєднує в собі передові алгоритми машинного навчання з мережами LSTM, що дозволяє їй точно ідентифікувати потенційні загрози або непередбачувані ситуації в домашньому середовищі. Це забезпечує високий рівень безпеки для користувачів. Крім програмного забезпечення, розробка також включає конструкторське виконання, що охоплює апаратне забезпечення та інтеграцію системи з різними компонентами домашнього господарства.

Завдяки своїй інноваційності, ефективності та комерційному потенціалу, ця система представляє значний інтерес для ринку «розумних» домашніх технологій, зокрема в сегменті домашньої безпеки. Вона демонструє як технічну новизну, так і практичну цінність, оскільки відповідає сучасним вимогам до зручності, ефективності та безпеки у повсякденному житті.

Для проведення комерційного та технологічного аудиту було залучено трьох незалежних експертів, які мають значний досвід роботи в галузі розробки програмного забезпечення, особливо у контексті штучного інтелекту та систем безпеки. Це забезпечило об'єктивність та професійність проведеного аудиту.

Оцінювання науково-технічного рівня та комерційного потенціалу розробки виконувалося на основі п’ятибальної системи оцінювання, яка включала 12 різних критеріїв. Ці критерії охоплювали широкий спектр аспектів, від технічної здійсненності концепції до ринкових перспектив та необхідності розробки регламентних документів. Такий підхід дозволив всебічно оцінити розробку,

враховуючи як її інноваційний потенціал, так і реальні можливості комерційного впровадження та успіху на ринку.

Застосування цієї методології забезпечило глибокий та мультидисциплінарний підхід до оцінки розробки, що дозволило отримати всебічне розуміння її сильних та слабких сторін, а також визначити перспективи її подальшого розвитку та комерціалізації.

У табл. 4.1 представлено результати проведеного оцінювання.

Таблиця 4.1 – Результати проведеного оцінювання розробки

№	Критерії	Експерт 1	Експерт 2	Експерт 3
1	Технічна здійсненність концепції	4	3	4
2	Аналоги на ринку	3	4	3
3	Ціна продукту	4	3	2
4	Технічні та споживчі властивості	4	3	4
5	Експлуатаційні витрати	3	4	4
6	Ринкові перспективи	4	3	4
7	Конкуренція на ринку	2	3	3
8	Практична здійсненність	3	4	3
9	Фінансові ресурси	4	3	2
10	Матеріали для реалізації ідеї	3	4	3
11	Термін реалізації ідеї	4	3	4
12	Необхідність розробки регламентних документів	3	2	4
Сума балів:		41	39	40
Середньоарифметична сума балів $СБ_с$:		40		

За результатами проведеного комерційного та технологічного аудиту, де середньоарифметична сума балів, розрахована на основі висновків експертів, склала 40 балів, було встановлено, що науково-технічний рівень та комерційний

потенціал розробки є високим. Такий високий рівень було досягнуто за рахунок кількох ключових факторів:

– інноваційність та унікальність розробки. Проект використовує передові технології у галузі штучного інтелекту, особливо LSTM мереж, що забезпечує новий підхід до систем безпеки в контексті розумних будинків. Це відкриває нові можливості для аналізу поведінкових патернів та прогнозування потенційних загроз;

– висока технічна здійсненність. Розробка продемонструвала високий рівень технічної здійсненності, включаючи успішні практичні випробування та експериментальні докази ефективності;

– конкурентоспроможність на ринку. Незважаючи на наявність аналогів на ринку, розробка вирізняється завдяки своїм унікальним характеристикам і технічним перевагам, що забезпечує їй високу конкурентоспроможність;

– ринкові перспективи. Проект має значний комерційний потенціал, оскільки він зосереджений на зростаючому попиті на інноваційні рішення у сфері домашньої безпеки;

– економічна ефективність. Розробка має потенціал для зменшення експлуатаційних витрат і пропонує більш ефективні рішення порівняно з існуючими аналогами, що робить її привабливою для споживачів.

Завдяки цим факторам, розробка не лише відповідає сучасним вимогам ринку, але й пропонує новаторські рішення, що значно підвищують рівень безпеки та комфорту в розумних будинках.

4.2 Розрахунок витрат на здійснення науково-дослідної роботи

4.2.1 Витрати на оплату праці

До статті «Витрати на оплату праці» належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим ви-конанням конкретної теми,

обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці, також будь-які види грошових і матеріальних доплат, які належать до елемента «Витрати на оплату праці» [42].

Витрати на основну заробітну плату дослідників ($З_о$) розраховують відповідно до посадових окладів працівників, за формулою:

$$З_о = \sum_{i=1}^{k} \frac{M_{ni} \cdot t_i}{T_p} \qquad (4.1)$$

де k – кількість посад дослідників, залучених до процесу досліджень;

M_{ni} – місячний посадовий оклад конкретного дослідника, грн;

t_i – кількість днів роботи конкретного дослідника, дн.;

T_p – середня кількість робочих днів в місяці, T_p =21 день.

За допомогою даних даної формули було проведено розрахунки витрати на заробітну плату дослідників та робітників, які представлено у табл. 4.2 та 4.3 відповідно.

Таблиця 4.2 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Кількість днів роботи	Витрати на заробітну плату, грн
Керівник проекту	18000	857,14	65	55714,1
Консультант з економіки	15000	714,29	1	714,29
Розробник	3600	174,42	65	11337,3
Всього:				67765,69

Витрати на основну заробітну плату робітників ($З_р$) за відповідними найменуваннями робіт розраховують за формулою:

$$З_р = \sum_{i=1}^{n} C_i \cdot t_i \qquad (4.2)$$

де C_i – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

t_i – час роботи робітника на виконання певної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду C_i можна визначити за формулою:

$$C_i = \frac{M_M \cdot K_i \cdot K_C}{T_p \cdot t_{зм}} \qquad (4.3)$$

де M_M – розмір прожиткового мінімуму працездатної особи або мінімальної місячної заробітної плати (залежно від діючого законодавства), грн;

K_i – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду;

K_c – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати. (табл. Б.1, додаток Б)

T_p – середня кількість робочих днів в місяці, приблизно T_p = 21…23 дні;

$t_{зм}$ – тривалість зміни, год.

Таблиця 4.3 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год ($t_{зм}$)	Розряд роботи	Тарифний коефіцієнт (K_i)	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
Монтаж обладнання	8	4	1,5	110	305,13
Прошивка контролера	2	3	1,35	150	274.62
Підключення датчиків	6	3	1,35	120	274.62
Всього:					854.37

Додаткова заробітна плата розраховується як 10 … 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$З_{дод} = (З_о + З_р) \cdot \frac{H_{дод}}{100\%} \qquad (4.4)$$

де $H_{дод}$ – норма нарахування додаткової заробітної плати (12%).

Отже, $З_{дод} = (67765{,}69 + 854{,}37) \cdot 0{,}12 = 8234{,}40$ (грн.)

4.2.2 Відрахування на соціальні заходи

До статті «Відрахування на соціальні заходи» належать відрахування внеску на загальнообов'язкове державне соціальне страхування та для здійснення заходів щодо соціального захисту населення (ЄСВ – єдиний соціальний внесок).

Нарахування на заробітну плату дослідників та робітників розраховується як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$З_н = (З_о + З_р + З_{дод}) \cdot \frac{Н_{зп}}{100\%} \qquad (4.5)$$

де $Н_{зп}$ – норма нарахування на заробітну плату.

Отже, $З_н = (67765{,}69 + 854{,}37 + 8234{,}40) \cdot 0{,}22 = 16907{,}98(\text{грн.})$.

4.2.3 Сировина та матеріали

До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби й предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень за прямим призначенням згідно з нормами їх витрачання, а також витрачені придбані напівфабрикати, що підлягають монтажу або виготовленню й додатковій обробці в цій організації, чи дослідні зразки, що виготовляються виробниками за документацією наукової організації.

Витрати на матеріали (М) у вартісному вираженні розраховуються окремо для кожного виду матеріалів за формулою:

$$M = \sum_{j=1}^{n} H_j \cdot Ц_j \cdot К_j - \sum_{j=1}^{n} В_j \cdot Ц_{вj} \qquad (4.6)$$

де Нj – норма витрат матеріалу j-го найменування, кг;

n – кількість видів матеріалів;

$Ц_j$ – вартість матеріалу j-го найменування, грн/кг;

$К_j$ – коефіцієнт транспортних витрат, ($К_j$ = 1,1 … 1,15);

$В_j$ – маса відходів j-го найменування, кг;

$Ц_{вj}$ – вартість відходів j-го найменування, грн/кг.

Проведені розрахунки були зведені до табл. 4.4.

Таблиця 4.4 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, кг	Величина відходів, кг	Ціна відходів, грн/кг	Вартість витраченого матеріалу, грн
Електричний кабель, мідний, 1.5 мм²	150.0	0.150	0.02	75.0	21.0
Хомут для кабелю, пластиковий, 100 мм	80.0	0.005	0.00	0.0	0.4
Гофрований захисний канал, ПВХ, 16 мм	60.0	0.100	0.02	30.0	5.4
Папір (формат А4)	110	1	0.00	0.0	110
Олов'яно-свинцевий припой	500	0.020	0.002	0.0	11
Всього					142,8

4.2.4 Розрахунок витрат на комплектуючі

Витрати на комплектуючі вироби ($К_в$), які використовують при дослідженні нового технічного рішення, розраховуються, згідно з їхньою номенклатурою, за формулою:

$$К_в = \sum_{j=1}^{n} Н_j \cdot Ц_j \cdot К_j \qquad (4.7)$$

де $Н_j$ – кількість комплектуючих j-го виду, шт.;

$Ц_j$ – покупна ціна комплектуючих j-го виду, грн;

$К_j$ – коефіцієнт транспортних витрат, ($К_j$ = 1,1 … 1,15).

Проведені розрахунки представлено у табл. 4.5.

Таблиця 4.5 – Витрати на комплектуючі

Найменування комплектуючих	Кількість, шт.	Ціна за штуку, грн	Сума, грн
Arduino UNO	1	1294	1294
Датчик pir-d203s	20	57	1140
Температурний датчик DHT22	4	227	908
Датчик Reed module KY-025	10	35	350
Фоторезистор KY-018	4	18	72
Мікрофонний модуль MAX9814	4	165	660
Сума:			4424

4.2.5 Спецустаткування для наукових (експериментальних) робіт

До статті «Спецустаткування для наукових (експериментальних) ро-біт» належать витрати на виготовлення та придбання спецустаткування, верстатів, пристроїв, інструментів, приладів, стендів, апаратів, механізмів, іншого спецобладнання, необхідного для проведення досліджень, також витрати на їх проектування, виготовлення, транспортування, монтаж та встановлення.

Вартість спецустаткування визначається за прейскурантом гуртових цін або за даними базових підприємств за відпускними і договірними ці-нами. До балансової вартості устаткування окрім прейскурантної вартості входять витрати на його транспортування і монтаж, тому ці витрати беруться додатково в розмірі 10…12% від вартості устаткування.

$$В_{спец} = \sum_{j=1}^{k} Ц_i \cdot С_{пр.i} \cdot К_i \qquad (4.8)$$

де $Ц_i$ – ціна придбання одиниці спецустаткування даного виду, марки, грн;

$С_{пр.i}$ – кількість одиниць устаткування відповідного найменування, які придбані для проведення досліджень, шт.;

K_i – коефіцієнт, що враховує доставку, монтаж, налагодження устатку-вання тощо, (K_i = 1,10…1,12);

k – кількість найменувань устаткування.

Отримані результати було зведено у табл. 4.6.

Таблиця 4.6 – Витрати на придбання спецустаткування по кожному виду

Найменування устаткування	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Паяльна станція	1	1200	1344
Мультиметр	1	500	560
Набір інструментів для монтажу	1	800	896
Всього			2800

4.2.6 Програмне забезпечення для наукових (експериментальних) робіт

Для розробки системи були використані безкоштовні іструменти:

– потужне ліцензійне інтегроване середовище розробки Visual Studio 2022 Community, яке розповсюджується безкоштовно для студентів, учасників проектів з відкритим кодом та окремих користувачів;

– Arduino IDE інтегроване середовище розробки для Windows, MacOS та Linux, розроблене на Сі та C++, призначене для створення та завантаження програм на Arduino-сумісні плати, а також на плати інших виробників.

Оскільки для розробки були використані безкоштовні інструменти, дані витрати можна не враховувати.

4.2.7 Амортизація обладнання, програмних засобів та приміщень

До цієї статті відносять амортизаційні відрахування по кожному виду обладнання, устаткування та інших приладів і пристроїв, а також програмного забезпе-чення для проведення науково-дослідної роботи, за його наявності в дослідній організації або на підприємстві.

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо можуть бути розраховані з використанням прямолінійного методу амортизації за формулою:

$$А_{обл} = \frac{Ц_б}{Т_в} \cdot \frac{t_{вик}}{12}, \qquad (4.9)$$

де $Ц_б$ – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{вик}$ – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

$Т_в$ – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

Проведені розрахунки представлені у табл. 4.7.

Таблиця 4.7 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Персональний комп’ютер	50000	4	3	3125
Операційна система	10000	3	3	833.33
Офісний пакет	5000	3	3	416.67
Паяльна станція	1344	5	3	67.20
Всього				4442.20

4.2.8 Паливо та енергія для науково-виробничих цілей

До цієї статті належать витрати на придбання у сторонніх підприємств, установ і організацій будь-якого палива, що витрачається з технологічною метою на проведення досліджень. Стаття формується у разі виконання енергоємних наукових досліджень за методом прямого внесення витрат і досягає значної питомої ваги у собівартості досліджень.

Витрати на силову електроенергію (B_e) розраховують за формулою:

$$\text{В}_\text{е} = \sum_{i=1}^{n} \frac{W_{yi} \cdot t_i \cdot \text{Ц}_\text{е} \cdot \text{К}_\text{впі}}{\eta_i} \qquad (4.10)$$

де W_{yi} – встановлена потужність обладнання на певному етапі розробки, кВт;

ti – тривалість роботи обладнання на етапі дослідження, год;

$Ц_е$ – вартість 1 кВт-години електроенергії, (7,5 грн.);

$К_{впі}$ – коефіцієнт, що враховує використання потужності, $К_{впі}$ <1;

η_i – коефіцієнт корисної дії обладнання, η_i <1.

Проведені розрахунки представлено у табл. 4.8.

Таблиця 4.8 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Персональний комп’ютер	0,45	400	3000
Система управління «Розумний дім»	0,1	400	300
Всього:			3300

4.2.9 Службові відрядження

Під час виконання роботи не було службових відряджень.

4.2.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації

Всі роботи проводились без допомоги сторонніх підприємств.

4.2.11 Інші витрати

Всі витрати були попередньо детально описані, додаткових витрат не було.

4.2.12 Накладні (загальновиробничі) витрати

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100…150% від суми основної заробітної плати дослідників та робітників за формулою:

$$В_{нзв} = (З_о + З_р) \cdot \frac{Н_{нзв}}{100\%}, \qquad (4.11)$$

де $Н_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати».

Згідно формули 4.6, $В_{нзв} = 16337{,}29 \cdot \frac{100\%}{100\%} = 16337{,}29$ (грн.).

Витрати на проведення науково-дослідної роботи розраховуються як сума всіх попередніх статей витрат за формулою:

$$В_{заг} = З_о + З_р + З_{дод} + З_н + М + К_в + В_{спец} + В_{прг} + А_{обл} + В_е + В_{св} + В_{сп} + І_в + В_{нзв} \qquad (4.12)$$

$$В_{заг} = 67765{,}69 + 854{,}37 + 8234{,}40 + 16907{,}98 + 142{,}8 + 4424 + 2800 + 4442.20 + 3300 = 108871{,}44 \text{ (грн.)}$$

Загальні витрати ЗВ на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються за формулою:

$$ЗВ = \frac{В_{заг}}{\eta} \qquad (4.13)$$

де η – коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи. Так, якщо науково-технічна розробка знаходиться на стадії: науково-дослідних робіт, то η=0,1; технічного проектування, то η =0,2; розробки конструкторської документації, то η=0,3; розробки технологій, то η=0,4; розробки

дослідного зразка, то η=0,5; розробки промислового зразка, то η=0,7; впровадження, то η=0,9.

Оскільки у даній роботі було розроблено концепцію інтелектуальної системи безпеки для розумних будинків на базі LSTM мереж, то η=0,4.

Отже, згідно 4.7:

$$ЗВ = \frac{1108871,44}{0,4} = 272178,60 \text{ (грн.)}$$

4.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

Розроблена інноваційна система «Розумний дім», яка базується на застосуванні передових технологій штучного інтелекту, зокрема мереж LSTM. Система розроблена з метою забезпечення підвищеної безпеки та ефективності в домашньому середовищі, інтегруючи розумні технології для автоматизації та оптимізації домашніх процесів. Основними функціями системи є точне виявлення вторгнень, адаптація до поведінки користувачів та персоналізація налаштувань.

Система «Розумний дім» розглядається не лише як окремий розумний пристрій, а як комплексна екосистема, здатна навчатися та адаптуватися до потреб і звичок користувача. Основна мета роботи - було створення продукту, який відрізняється високим рівнем інноваційності, безпеки та комфорту, що відповідає сучасним вимогам та трендам ринку домашньої автоматизації.

Передбачається, що впровадження такої системи на ринок домашньої автоматизації зумовить зростання інтересу з боку потенційних споживачів. Завдяки покращенню функціональності та зручності, система «Розумний дім» має усі шанси стати вибором для споживачів, що цінують інтегровані рішення високої якості. Продукт пропонує не тільки підвищення рівня безпеки житла, але й забезпечує додаткові можливості для автоматизації та контролю домашнього середовища.

Для всіх наведених випадків можливе збільшення чистого прибутку у потенційного інвестора $\Delta П_i$ для кожного із років, протягом яких очікується

отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховується за формулою:

$$\Delta П_і = (\pm \Delta Ц_о \cdot N + Ц_о \cdot \Delta N)_i . \lambda \cdot \rho (1 - \frac{\vartheta}{100}), \qquad (4.14)$$

де $\pm \Delta Ц_о$– зміна основного якісного показника від впровадження результатів науково-технічної розробки в аналізованому році. Зазвичай, таким показником може бути зміна ціни реалізації одиниці нової розробки в аналізованому році (відносно року до впровадження цієї розробки); $\pm \Delta Ц_о$ може мати як додатне, так і від'ємне значення (від'ємне – при зниженні ціни відносно року до впровадження цієї розробки, додатне – при зростанні ціни);

N– основний кількісний показник, який визначає величину попиту на аналогічні чи подібні розробки у році до впровадження результатів нової науково-технічної розробки;

$Ц_о$ – основний якісний показник, який визначає ціну реалізації нової науково-технічної розробки в аналізованому році, $Ц_о = Ц_б \pm \Delta Ц_о$;

$Ц_б$ – основний якісний показник, який визначає ціну реалізації існуючої (базової) науково-технічної розробки у році до впровадження результатів;

ΔN – зміна основного кількісного показника від впровадження результатів науково-технічної розробки в аналізованому році. Зазвичай таким пока-зником може бути зростання попиту на науково-технічну розробку в аналізованому році (відносно року до впровадження цієї розробки);

λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2023 році ставка податку на додану вартість становить 20%, а коефіцієнт λ=0,8333;

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту (послуги). Рекомендується брати ρ=0,2...0,5;

ϑ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2023 році ϑ=18%.

Для розрахунку $\Delta П_і$ було використано наступні показники:

– $\pm\Delta Ц_o$ Зміна ціни становить +500 грн. Це виправдано підвищенням якості та функціональності продукту;

– N. Для початку використано 15000 одиниць як базовий показник попиту на подібні системи;

– $Ц_o$ ціна реалізації становить 30000 грн за одиницю;

– ΔN зростання попиту може в середньому становити 200 одиниць кожного року, що відображає зростаючий інтерес до продукту;

– λ=0,8333;

– ρ=0,3;

– ϑ=18%.

Згідно формули 4.14, отримаємо:

$$\Delta П_1 = (200 \cdot 15000 + 30500 \cdot 200) \cdot 0{,}8333 \cdot 0{,}3(1 - 0{,}18) = (3000000 + 6100000) \cdot 0{,}2 = 1820000 \text{ грн.}$$

$$\Delta П_2 = (200 \cdot 15000 + 30500 \cdot (200 + 200)) \cdot 0{,}8333 \cdot 0{,}3(1 - 0{,}18) = (3000000 + 12200000) \cdot 0{,}2 = 3040000 \text{ грн.}$$

$$\Delta П_3 = (200 \cdot 15000 + 30500 \cdot (200 + 200 + 200)) \cdot 0{,}8333 \cdot 0{,}3(1 - 0{,}18) = (3000000 + 18300000) \cdot 0{,}2 = 4260000 \text{ грн.}$$

Далі розраховують приведену вартість збільшення всіх чистих прибутків ПП, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{i=1}^{T} \frac{\Delta П_i}{(1+\tau)^t} \qquad (4.15)$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

T – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, τ=0,05…0,15;

t – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

Для цього були взяті такі значення: Т=3 (три роки), τ= 0.1 (10%)

$$ПП = \frac{1820000}{(1+0,1)^3} + \frac{3040000}{(1+0,1)^3} + \frac{4260000}{(1+0,1)^3} = 1367392,94 + 2283996,99 + 3200601,05 = 6851990,98 \text{ грн.}$$

Далі розраховують величину початкових інвестицій PV, які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{інв} \cdot ЗВ \qquad (4.16)$$

де $k_{інв}$– коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай $k_{інв}$=2...5, але може бути і більшим;

ЗВ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

Згідно 4.16 $PV = 3 * 272178,60 = 816535,80$ (грн.)

Тоді абсолютний економічний ефект $E_{абс}$ або чистий приведений дохід (NPV, Net Present Value) для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$Е_{абс} = ПП - PV, \qquad (4.17)$$

де ПП – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, грн;

PV – теперішня вартість початкових інвестицій, грн.

Якщо величина $Е_{абс}$буде мати велике додатне значення, то це може сві-дчити про потенційну зацікавленість інвесторів у впровадженні та комерціалізації цієї науково-технічної розробки.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність $E_в$ або показник внутрішньої норми дохідності (IRR, Internal Rate of Return) вкладених інвестицій.

Згідно 4.12 $\text{Е}_{\text{абс}} = 6851990{,}98 - 816535{,}80 = 6035455{,}18$ (грн.)

Внутрішня економічна дохідність інвестицій $E_в$, які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки, розраховується за формулою:

$$\text{Е}_\text{в} = \sqrt[\text{Тж}]{1 + \frac{\text{Е}_{\text{абс}}}{PV}} - 1, \qquad (4.18)$$

де $E_{абс}$ – абсолютний економічний ефект вкладених інвестицій, грн;

PV – теперішня вартість початкових інвестицій, грн;

$\text{Т}_\text{ж}$ – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримування позитивних результатів від її впровадження, роки.

Далі визначають бар'єрну ставку дисконтування $\tau_{мін}$, тобто мінімальну внутрішню економічну дохідність інвестицій, нижче якої кошти у впровадження науково-технічної розробки та її комерціалізацію вкладатися не будуть.

$\text{Т}_\text{ж}$ = 3 (3 роки).

Згідно 4.13 розрахуємо $\text{Е}_\text{в} = \sqrt[3]{1 + \frac{6035455{,}18}{816535{,}80}} - 1 = 2 - 1 = 1$

Мінімальна внутрішня економічна дохідність вкладених інвестицій мінтвизначається за формулою:

$$\tau_{\text{мін}} = d + f \qquad (4.19)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2023 році в Україні d=0,09...0,12;

f – показник, що характеризує ризикованість вкладення інвестицій; зазвичай величина f=0,05...0,5, але може бути і значно вищою.

Якщо величина $\text{Е}_\text{в} > \tau_{\text{мін}}$, то потенційний інвестор може бути зацікавлений у фінансуванні впровадження науково-технічної розробки та виве-денні її на ринок, тобто в її комерціалізації.

Візьмемо d=0,09 та *f*=0,5

Згідно 4.19 розрахуємо $\tau_{мін} = 0{,}09 + 0{,}1 = 0{,}19$

Далі розраховуємо період окупності інвестицій $T_{ок}$ (DPP, Discounted Payback Period), які можуть бути вкладені потенційним інвестором у впро-вадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_в}, \qquad (4.20)$$

де $E_в$– внутрішня економічна дохідність вкладених інвестицій.

Якщо $T_{ок}$< 3-х років, то це свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження цієї розробки та виведення її на ринок.

Згідно 4.15 $T_{ок} = \frac{1}{1} = 1$, що свідчить про термін окупності продукту за 1 рік після старту продаж.

4.4 Висновок до розділу

В даному розділі проведено розрахунок витрат на розробку проектного рішення. Згідно проведених розрахунків було підтверджено комерційну привабливість науково-технічної розробки, що у свою чергу може спонукати потенційних інвесторів профінансувати впровадження цієї розробки та виведення її на ринок.

ВИСНОВКИ

1. Проаналізовано історію та концепцію розумних будинків. З'ясовано, що централізовані та децентралізовані архітектури «Розумний дім» мають свої переваги та недоліки, важливо вибирати оптимальну структуру залежно від конкретних потреб та умов експлуатації.

2. Розроблено математичну модель прогнозування поведінкових шаблонів користувачів системи «Розумний дім» на основі мереж LSTM, що дозволяє підвищити точність та швидкість реагування системи на потенційні загрози.

3. Проведено аналіз потенційних загроз та ризиків у системах «Розумний дім», виявлено ключові вразливі місця та методи їх нейтралізації.

4. Експериментально досліджено ефективність пропонованого рішення. За допомогою реалізованих алгоритмів ідентифікації та класифікації потенційних загроз на основі LSTM досягнуто значного покращення у здатності системи розпізнавати й реагувати на різноманітні сценарії.

5. Виявлено шляхи подальшого вдосконалення та розвитку системи, включаючи інтеграцію з іншими компонентами «Розумного дому» та покращення алгоритмів LSTM для більш точного аналізу даних.

6. Проведено економічний аналіз науково-дослідної роботи, встановлено, що розробка є комерційно привабливою та має потенціал для впровадження на ринку, що може привернути потенційних інвесторів.

Рекомендації щодо наукового та практичного використання одержаних результатів дослідження:

- інтеграція розробленої моделі у існуючі системи «Розумний дім». Модель, заснована на LSTM, може бути впроваджена в сучасні системи для підвищення їх ефективності у прогнозуванні та запобіганні потенційних загроз;
- розвиток алгоритмів машинного навчання для подальшого покращення безпеки. На основі отриманих даних та результатів експериментів можна розробляти більш складні та точні алгоритми, які зможуть краще адаптуватися до змінних умов та нових типів загроз;

- навчання персоналу та користувачів системи. Важливо організувати тренінги та курси для технічного персоналу та кінцевих користувачів, щоб вони розуміли принципи роботи та можливості нової системи безпеки;
- проведення подальших досліджень у цій області. Рекомендується продовжити дослідження в області використання мереж LSTM для підвищення безпеки систем розумного дому, що дозволить розробити ще більш ефективні методи захисту;
- співпраця з розробниками та виробниками обладнання. Налагодження взаємодії з компаніями, що виробляють компоненти для розумних домів, для інтеграції розроблених рішень в їх продукцію;
- залучення інвестицій для комерціалізації проекту. Підготовка та представлення проекту потенційним інвесторам, щоб забезпечити фінансування для його подальшої розробки та впровадження на ринку.

Зформовані рекомендації допоможуть використати результати дослідження для практичного застосування та подальшого розвитку в галузі систем безпеки «Розумний дім».

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. *Long short-term memory* (LSTM) : веб-сайт. URL: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

2. «Горизонт 2020» (Horizon 2020) : веб-сайт. URL: https://eu-ua.org/horizon-2020/

3. Що таке «розумний будинок» і навіщо він потрібен? : веб-сайт. URL: https://stylus.ua/uk/articles/528.html

4. Що таке розумний будинок? Все що потрібно знати про систему Розумний Дім : веб-сайт. URL: https://bron.ua/article/schotake-rozumnij-budinok-vse-scho-potrbno-znati-pro-sistemu-rozumnij-dm/5/

5. Patrascu M. Integrating Services and Agents for Control and Monitoring: Managing Emergencies in Smart Buildings. Service Orientation in Holonic and MultiAgent Manufacturing and Robotics. / Patrascu., 2014. – 544 с

6. Granzer W. P. Security in Building Automation Systems / Wolfgang Praus Granzer. Munich: Appress, 2018. – 578 с.

7. Технологія розумного будинку: як AI створює простір, комфортний для життя : веб-сайт. URL: https://www.everest.ua/tehnologiya-rozumnogo-budynku-yak-ai-stvoryuye-prostirkomfortnyj-dlya-zhyttya/

8. RL-IoT: Reinforcement Learning to Interact with IoT Devices : веб-сайт. URL: https://arxiv.org/abs/2105.00884

9. Q-Learning : веб-сайт. URL: https://www.simplilearn.com/tutorials/machine-learning-tutorial/what-is-q-learning

10. Discoverer : веб-сайт. URL: https://azuremarketplace.microsoft.com/en-us/marketplace/consulting-services/ntt_data.it_iot_discovery

11. Future research challenges in wireless sensor and actuator networks targeting industrial automation: веб-сайт. URL: https://ieeexplore.ieee.org/abstract/document/6034912.

12. . Дужак І. О. «Розумний будинок» // Автоматизація технол. і бізнес-процесів. – 2013. – № 13-14. – С. 31-33. – URL: http://journals.uran.ua/atbp/article/download/32920/29533

13. «Розумний будинок» : бібліографічний покажчик / КЗ «ЗОУНБ» ЗОР, Від. наук. інформації та бібліографії ; [уклад. М. Маслова]. – Запоріжжя : [ЗОУНБ], 2021. – 76 с.

14. Струков В. Система інтелектуальної автоматизації «Розумний будинок» / Струков В., Сабо А. Г. // Проблеми механізації та електрифікації технологічних процесів : матеріали VI Всеукр. наук.-техн. Інтернет-конф. молод. учених, магістрантів та студентів за підсумками наук. дослідж. 2018 р. – Мелітополь, 2019. – Вип. VI. – С. 66-67. – URL: https://u.to/NVZEGw

15. Кучеренко А. О. «Бездротові передавачі камер відеоспостереження в інформаційно-охоронній системі «Розумний будинок» // Science Online. International Electronic Scientific Journal. – 2018. – № 7. – 5 с. – URL: https://u.to/03VGGw

16. Топольський А. І. Проектування програмного забезпечення системи управління освітленням / А. І. Топольський, О. О. Ковалюк // L Наук.-техн. конф. ф-ту комп'ютер. систем і автоматики (18.03.2021) / ВНТУ, Ф-т комп'ютер. систем і автоматики. – Вінниця, 2021. – 3 с. – URL: https://u.to/1A5HGw

17. Король З. З. Аналіз вразливостей системи захисту в об'єктах типу «Розумний Будинок» : [магістер. робота] / Король З. З. ; МОН України, Запоріз. нац. техн. ун-т. – Запоріжжя, 2018. – 91 с.

18. Дерев'янко Ю. В. Дослідження можливостей «інтелектуального будинку» / Ю. В. Дерев'янко, О. Л. Краснікова // Право і Безпека. – 2010. – № 1. – С. 223-226.

19. What is LSTM? Introduction to Long Short-Term Memory : веб-сайт. URL: https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/

20. User Activity Sequence Prediction in Smart Homes using Multi-Layer Long Short-Term Memory Networks : веб-сайт. URL: https://www.sciencedirect.com/science/article/pii/S2405896321023302

21. Загрози і вразливості безпеки розумного будинку : веб-сайт. URL: https://naukam.triada.in.ua/index.php/konferentsiji/70-tridtsyat-dev-yata-vseukrajinska-

praktichno-piznavalna-internet-konferentsiya/933-zagrozi-i-vrazlivosti-bezpeki-rozumnogo-budinku

22. Mouaatamid O., Lahmer M., Belkasmi M. Internet of Things Security: Layered classification of attacks and possible Countermeasures. Electronic Journal Of Information Technology, 2016, 9

23. Савченко К. В., Войтович О. П. Структурна схема системи захисту розумного будинку // Матеріали конференції XLVI Науково – технічна конференція факультету інформаційних технологій та комп'ютерної інженерії(2017) : веб-сайт. URL: https://conferences.vntu.edu.ua/index.php/all – fitki/all – fitki – 2017/paper/view/2736

24. Основні небезпеки приладів у складі розумного будинку. Із чим стикаються споживачі? : веб-сайт. URL: https://www.itsec.ru/articles/osnovnye-opasnosti-ustrojstv-vsostave-umnogo-doma-s-chem-stalkivayutsya-potrebiteli

25. Роль штучного інтелекту в сучасних системах безпеки : веб-сайт. URL: https://worldvision.com.ua/rol-iskusstvennogo-intellekta-v-sovremennykh-sistemakh-bezopasnosti/

26. Convolutional Neural Networks – CNN : веб-сайт. URL: https://www.mathworks.com/discovery/convolutional-neural-network-matlab.html

27. Random Forest : веб-сайт. URL: https://www.ibm.com/topics/random-forest

28. Gradient Boosting : веб-сайт. URL: https://www.analyticsvidhya.com/blog/2021/09/gradient-boosting-algorithm-a-complete-guide-for-beginners/

29. Home Smart IoT Home : веб-сайт. URL: https://www.toptal.com/designers/interactive/smart-home-domestic-internet-of-things

30. Natural Language Processing with Deep Learning. URL: https://web.stanford.edu/class/cs224n/readings/cs224n-2019-notes05-LM_RNN.pdf.

31. Creating a Business Logic Layer : веб-сайт. URL: https://learn.microsoft.com/en-us/aspnet/web-forms/overview/data-access/introduction/creating-a-business-logic-layer-cs

32. Creating a Data Access Layer : веб-сайт. URL:https://learn.microsoft.com/en-us/aspnet/web-forms/overview/data-access/introduction/creating-a-data-access-layer-cs

33. Global Cybersecurity Index (GCI) 2018 [Електронний ресурс]. – Режим доступу : https://www.itu.int/en/ITU-D/Cybersecurity/Documents/draft-18-00706_Global-Cybersecurity-IndexEV5_print_2.pdf .

34. Arduino UNO : веб-сайт. URL: https://doc.arduino.ua/ru/hardware/Uno

35. Датчик PIR (Passive Infrared) D203S : веб-сайт. URL: https://electronica.in.ua/ua/p1554094886-pir-datchik-d203s.html

36. Температурний датчик DHT22 : веб-сайт. URL: https://diylab.com.ua/p520038120-datchik-temperaturi-vologosti.html

37. KY-025 Reed module: веб-сайт. URL: https://datasheetspdf.com/pdf-file/1402036/Joy-IT/KY-025/1.

38. KY-018 Photoresistor module. URL: https://datasheetspdf.com/pdf-file/1402029/Joy-IT/KY-018/1.

39. Microphone Amplifier with AGC and Low-Noise Microphone Bias : веб-сайт. URL: https://www.analog.com/media/en/technical-documentation/data-sheets/max9814.pdf.

40. Microphone Amplifier with AGC and Low-Noise Microphone Bias: веб-сайт. URL: https://www.digchip.com/datasheets/1136701-max9814.html.

41. Dataset for Smart Home: веб-сайт. URL: https://www.kaggle.com/datasets/shegguy87/dataset-for-smart-home.

42. В. О. Козловський, О. Й. Лесько, В. В. Кавецький. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт – Вінниця : ВНТУ, 2021. – 42 с.

ДОДАТКИ

Вінницький національний технічний університет
Факультет менеджменту та інформаційної безпеки
Кафедра менеджменту та безпеки інформаційних систем

ЗАТВЕРДЖУЮ
Голова секції "Управління інформаційною безпекою" кафедри МБІС
д.т.н., професор
_____ Юрій ЯРЕМЧУК
"20" вересня 2023 р.

ТЕХНІЧНЕ ЗАВДАННЯ

до магістерської кваліфікаційної роботи на тему:
Підвищення захищеності системи розумний дім засобами штучного інтелекту на основі мережі LSTM

08-72.МКР.000.00.115.ТЗ

Керівник магістерської кваліфікаційної роботи
к.т.н., доцент
_____ Сачанюк-Кавецька Н. В.

Вінниця – 2023 р.

1. Найменування та область застосування

Розробка інтелектуальної системи безпеки для "Розумних будинків", здатної протистояти сучасним кіберзагрозам за допомогою технологій штучного інтелекту на основі LSTM мереж. Область застосування: система призначена для використання у "Розумних будинках" для ідентифікації та попередження несанкціонованого доступу та проникнення. Це досягається шляхом аналізу поведінкових паттернів мешканців та виявлення незвичних дій або втручань, що можуть вказувати на присутність сторонніх осіб або загрози безпеці. Система використовує передові технології штучного інтелекту для точного виявлення таких ситуацій.

2. Підстава для розробки

Розробка виконується на основі наказу ректора ВНТУ №203 від 14. 09. 2022 р.

3. Мета та призначення розробки

3.1 Мета розробки: створення системи безпеки, що базується на поведінкових паттернах мешканців для захисту "Розумних будинків" від несанкціонованого проникнення та інших загроз.

3.2 Призначення: розроблена система буде використовувати алгоритми штучного інтелекту для точного аналізу поведінки мешканців та виявлення незвичайних або підозрілих дій, спрямованих на забезпечення безпеки дому та його мешканців.

4. Джерела розробки

4.1. Long short-term memory (LSTM) : веб-сайт. URL: https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

4.2. Patrascu M. Integrating Services and Agents for Control and Monitoring: Managing Emergencies in Smart Buildings. Service Orientation in Holonic and MultiAgent Manufacturing and Robotics. / Patrascu., 2014. – 544 с.

4.3. Granzer W. P. Security in Building Automation Systems / Wolfgang Praus Granzer. Munich: Appress, 2018. – 578 с.

4.4. Natural Language Processing with Deep Learning. URL: https://web.stanford.edu/class/cs224n/readings/cs224n-2019-notes05-LM_RNN.pdf.

5. Вимоги до програми

5.1 Вимоги до функціональних характеристик:

5.1.1 Система повинна мати інтуїтивно зрозумілий та зручний інтерфейс користувача.

5.1.2 Реалізація системи повинна бути можлива без використання ліцензійних програмних додатків.

5.1.3 Система повинна забезпечувати ефективний механізм автентифікації користувачів.

5.2 Вимоги до надійності:

5.2.1 Система повинна працювати стабільно і безперебійно, з відповідним повідомленням у разі критичних помилок.

5.2.2 Бази даних повинні мати механізм автоматичного резервного копіювання.

5.2.3 Система повинна безперервно виконувати свої функції.

5.3 Вимоги до складу і параметрів технічних засобів:

– процесор: Intel Core i5 або аналогічний;
– оперативна пам'ять: 4 GB;
– жорсткий диск: 256 GB SSD;
– операційна система: сумісність з Windows 10 або новішими версіями, MacOS, а також підтримка Linux дистрибутивів.

6. Вимоги до програмної документації

Після запуску системі, на екрані з'явиться вікно, в якому користувачу буде запропоновано ввести своє ім'я та пароль. Це необхідно для ідентифікації користувача та забезпечення безпеки його особистих даних. Користувач повинен ввести своє ім'я та пароль у відповідні поля вікна та натиснути кнопку "Увійти". Після цього програмне забезпечення перевірить введені дані та перенаправить користувача на головний екран.

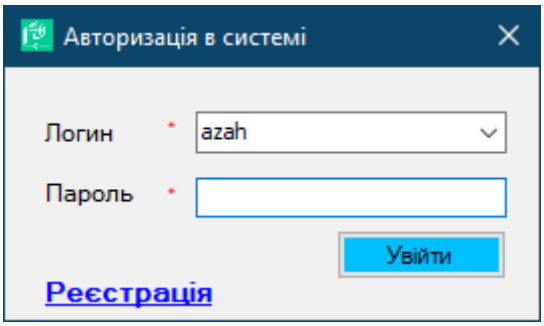


Рисунок 1 – Автентифікація користувача в системі

Система також дозволяє здійснювати реєстрацію нового користувача, якщо у нього немає облікового запису. Для цього йому необхідно натиснути кнопку «Реєстрація», після чого буде виведено вікно, що зображено на рис. 2.

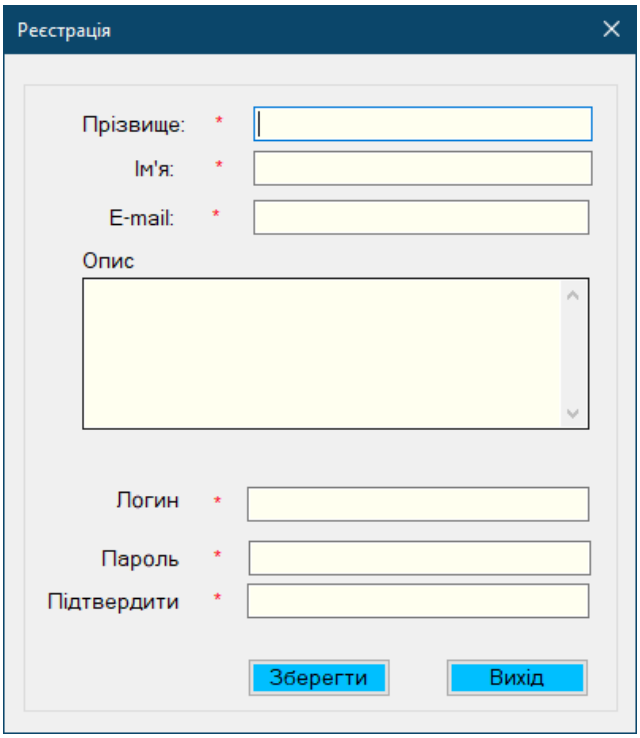


Рисунок 2 – Форма реєстрації користувачів

Якщо у програмному забезпеченні ще не зареєстровано жодного користувача, то при реєстрації першому користувачу автоматично буде присвоєна роль викладача, іншим користувачам буде присвоєно ролі студентів.

Для швидкого пошуку імені в сервісі "Онлайн-бібліотека освітніх ресурсів" можна скористатись випадаючим списком, який містить доступні імена користувачів. Крім того, при введенні імені з клавіатури програмне забезпечення автоматично поставить його у верхню частину списку, що спрощує інтерфейс і дозволяє швидше знайти потрібне ім'я.

Після успішної автентифікації користувача у систему буде відкриватиметься головне вікно (рис. 3). Це вікно містить основне меню, що складається з доступних опцій та функцій програмного забезпечення.

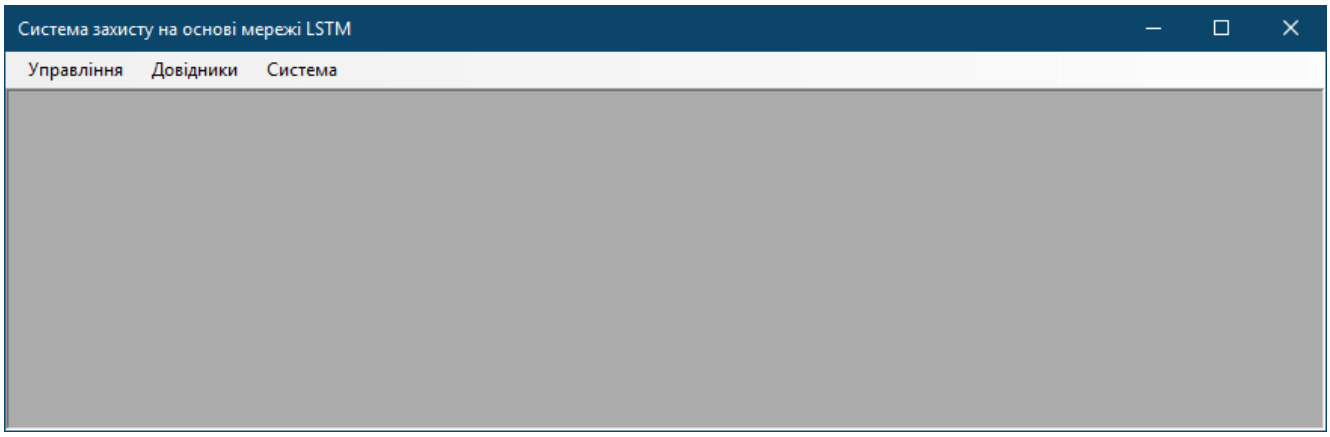


Рисунок 3 – Головне меню програми

У системі реалізовано дві ролі: системного адміністратора та користувача. Роль системного адміністратора дозволяє керувати обліковими записами та переглядати події системи.

Роль користувача дозволяє створювати категорії загроз, тренувати моделі для виявлення загроз, а також проводити тестування моделей.

Для створення категорії загроз, користувачу системи необхідно перейти по меню «Довідники» –> «Категорії», після чого відкриється вікно, що представлено на рис. 4.

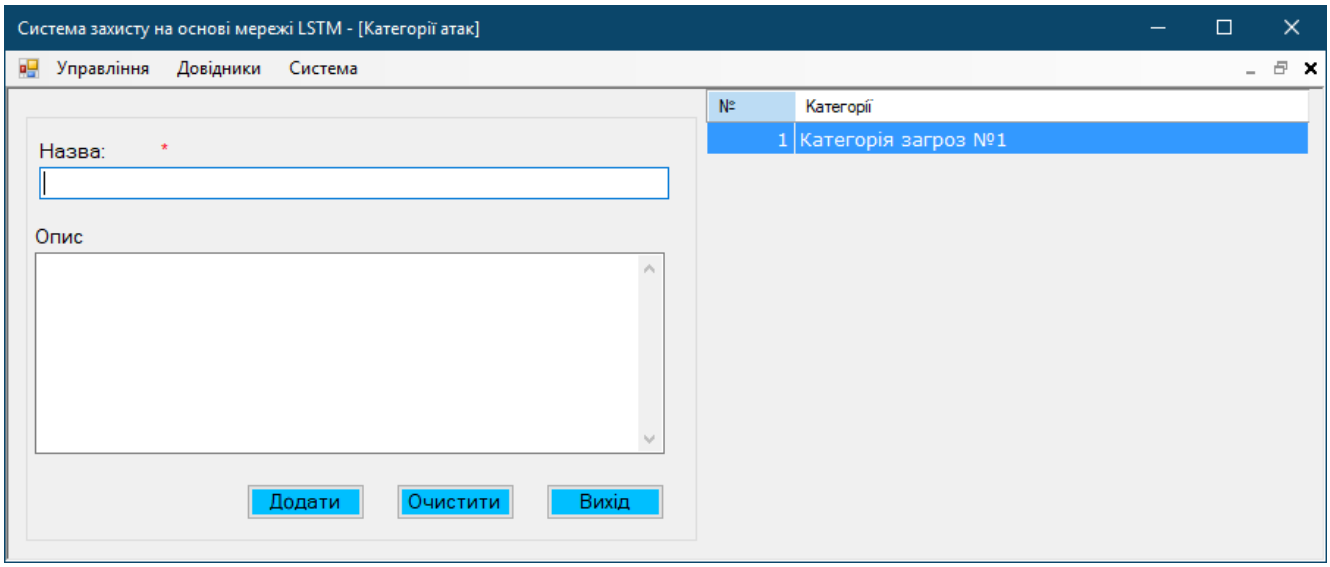


Рисунок 4 – Створення категорії навчання моделі

Для тренування нової моделі згідно підготовленого датасету, потрібно здійснити перехід по меню програми «Довідники» –> «Тренування моделей». Після чого натиснувши кнопку «Відкрити» обрати через діалогове вікно підготовлений датасет. Після вибору файлу процес начання моделі бузе проводитись автоматично. На початку буде виведено повідомлення «Training the model...», а після навчання буде виведено значення метрик (рис. 5).

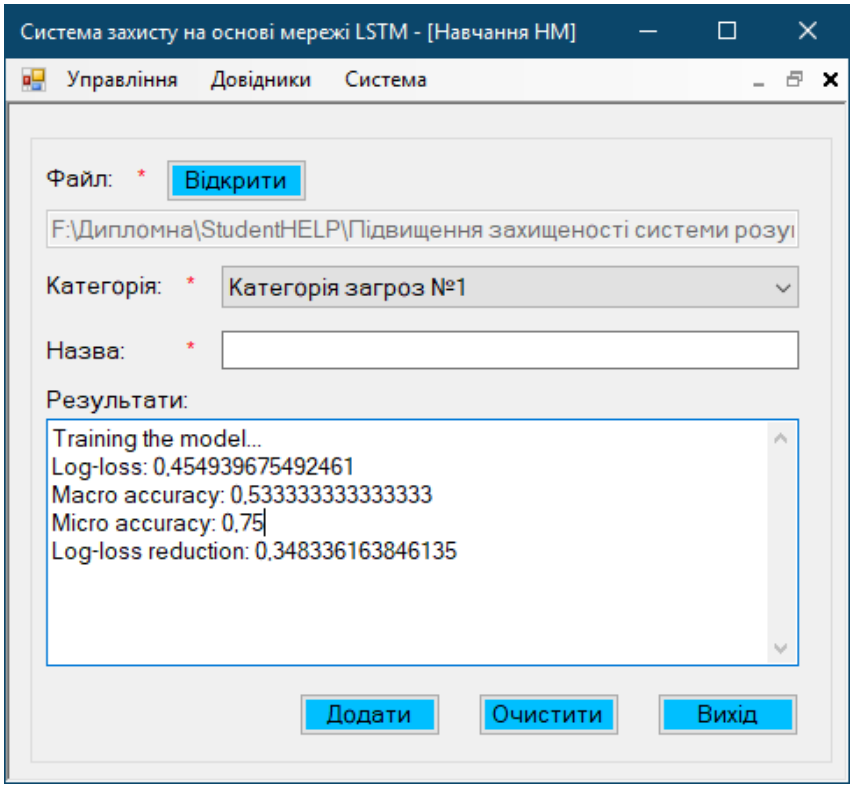


Рисунок 5 – Результат навчання НМ

Після цього навчену модель можна зберегти у системі вказавши назву моделі та обравши категорію (рис. 6).

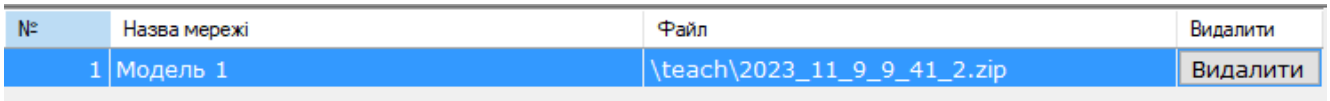


Рисунок 6 – Збереження моделі НМ у системі

Для проведення тестування моделі користувачу потрібно перейти по меню програми «Управління» –> «Тестування моделей» та обравши відповідну категорію запустити генератор загроз (рис. 7).

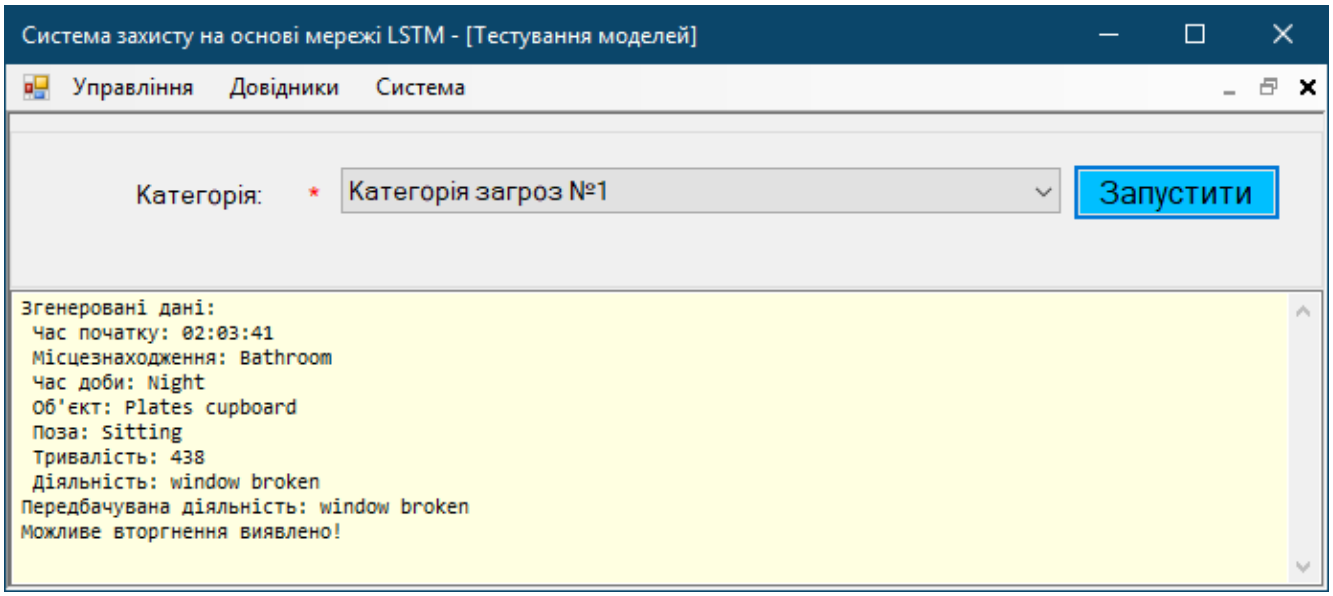


Рисунок 7 – Тестування моделей з допомогою генератора загроз

Управління обліковими записами є важливою частиною багатьох систем, і вимагає обережності та відповідальності в їх використанні. Користувач із роллю системного адміністратора має доступ до управління обліковими записами системи (рис. 8).

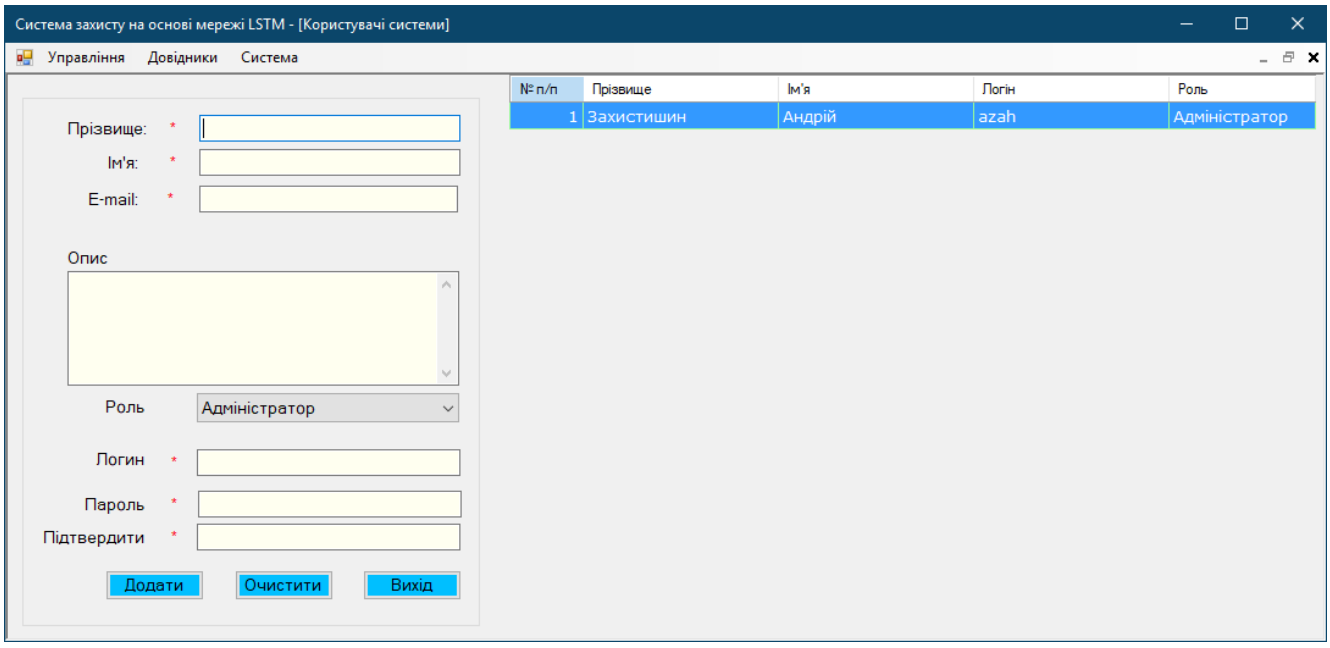


Рисунок 8 – Управління обліковими записами

Таким чином адміністратори можуть мати можливість створювати, редагувати та видаляти облікові записи користувачів, які мають доступ до даної системи. Також за допомогою даної ролі можна змінювати ролі системи. Це

дозволяє адміністраторам забезпечувати безпеку даних та контролювати доступ користувачів до ресурсів (рис. 9).

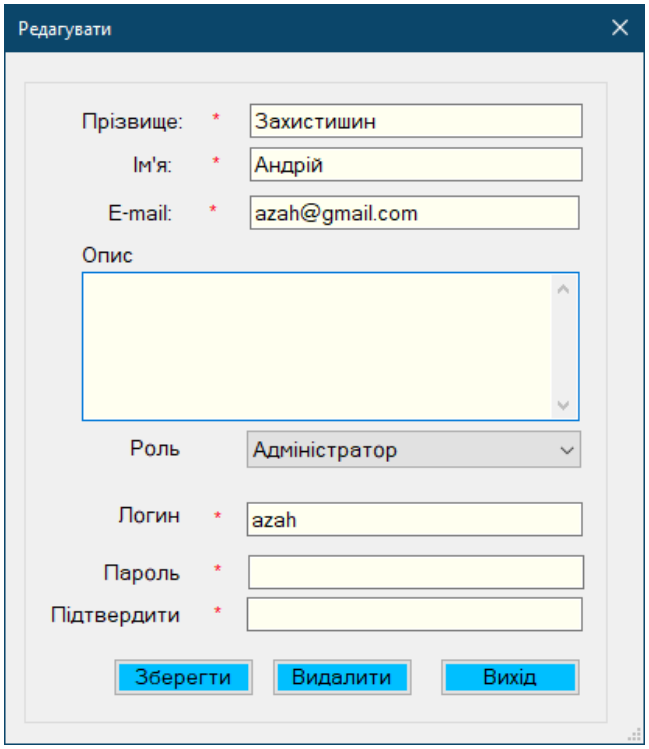


Рисунок 9 – Редагування даних користувача системи

Якщо ж поля «Пароль» та «Підтвердити» не співпадають, система виведе повідомлення про це (рис. 10).

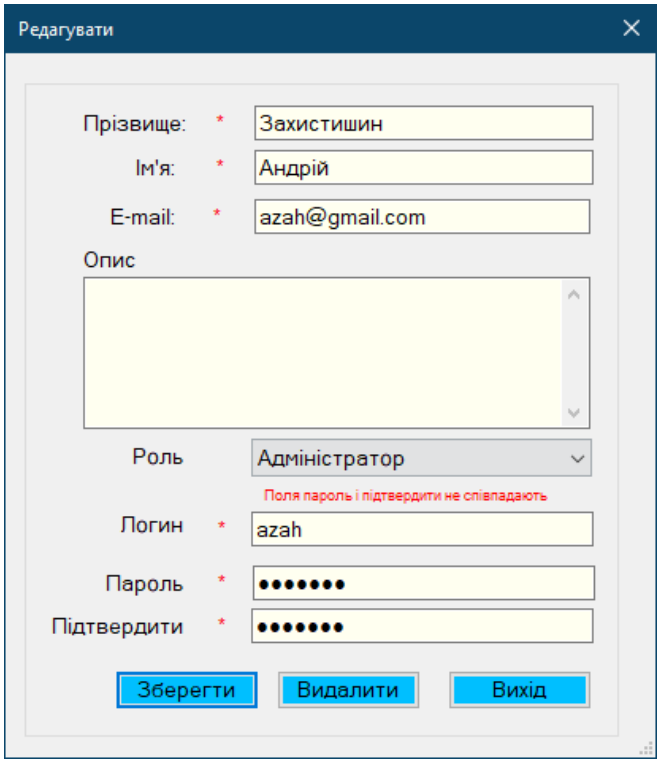
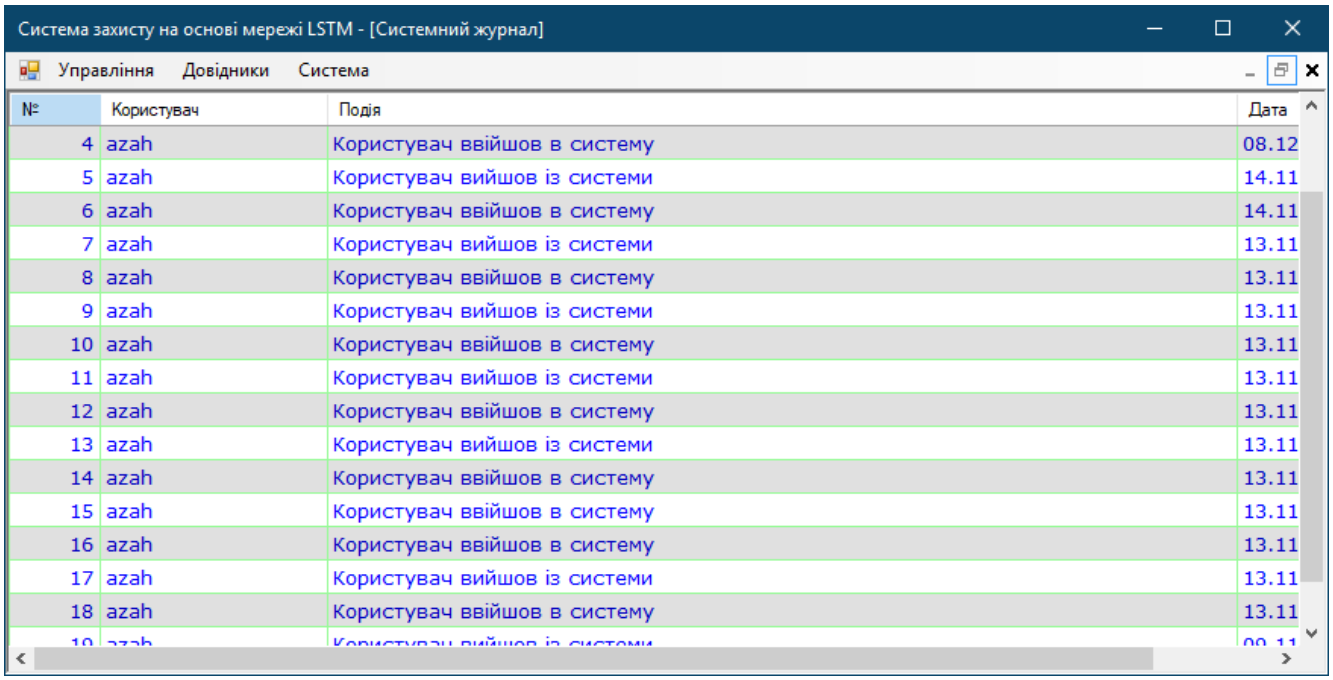


Рисунок 10 – Пароль» та «Підтвердити» не співпадають

В системі можна відстежувати активність користувачів та їх дії завдяки обліковому запису з правами адміністратора. Це можливо зробити за допомогою функції "Події системи", яка знаходиться в меню "Система" (рис. 11).



№	Користувач	Подія	Дата
4	azah	Користувач ввійшов в систему	08.12
5	azah	Користувач вийшов із системи	14.11
6	azah	Користувач ввійшов в систему	14.11
7	azah	Користувач вийшов із системи	13.11
8	azah	Користувач ввійшов в систему	13.11
9	azah	Користувач вийшов із системи	13.11
10	azah	Користувач ввійшов в систему	13.11
11	azah	Користувач вийшов із системи	13.11
12	azah	Користувач ввійшов в систему	13.11
13	azah	Користувач вийшов із системи	13.11
14	azah	Користувач ввійшов в систему	13.11
15	azah	Користувач ввійшов в систему	13.11
16	azah	Користувач ввійшов в систему	13.11
17	azah	Користувач вийшов із системи	13.11
18	azah	Користувач ввійшов в систему	13.11

Рисунок 11 – Події системи

Після відкриття вікна "Системний журнал" користувач може переглянути список всіх подій, що сталися в системі. Це можуть бути дії користувачів, пов'язані з доступом до ресурсів, редагуванням даних тощо.

Інформація в системному журналі може бути корисною для визначення проблем з безпекою та знаходження вразливостей в системі. Також, вона може бути використана для відстеження дій користувачів та їх контролю.

Користуючись меню «Персоналізація» можна змінити облікові дані.

Для виходу із програми необхідно перейти по меню «Управління» –> «Вихід».

7. Вимоги до технічного захисту інформації:

7.1 Необхідно впровадити заходи для запобігання несанкціонованого доступу та використання розроблюваної системи безпеки.

7.2 Система повинна гарантувати, що доступ до інформаційних ресурсів можливий лише для авторизованих користувачів.

7.3 Використання надійного шифрування для забезпечення конфіденційності даних, переданих та зберіганих в системі.

7.4 Реалізація механізмів моніторингу та аудиту для виявлення та реагування на підозрілі дії або спроби порушення.

8. Техніко-економічні показники:

8.1 Очікувана цінність та користь від впровадження проекту має значно перевищувати його вартість розробки та експлуатації.

8.2 Проект має бути реалізований з урахуванням потреб широкого кола користувачів, забезпечуючи легкість у використанні та доступність.

9. Стадії та етапи розробки

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Початок	Закінчення
1	Визначення напрямку магістерської роботи, формулювання теми	20.09.2023	25.09.2023
2	Аналіз предметної області обраної теми	26.09.2023	28.09.2023
3	Апробація отриманих результатів	29.09.2023	02.10.2023
4	Розробка алгоритму роботи	03.10.2023	17.10.2023
5	Написання магістерської роботи на основі розробленої теми	18.10.2023	10.11.2023
6	Розробка економічної частини	11.11.2023	23.11.2023
7	Передзахист магістерської кваліфікаційної роботи	24.11.2023	25.11.2023
8	Виправлення, уточнення, корегування магістерської кваліфікаційної роботи	26.11.2023	30.11.2023
9	Захист магістерської кваліфікаційної роботи	15.12.2023	15.12.2023

10. Порядок контролю та прийому

10.1 До приймання магістерської кваліфікаційної роботи надається:

– ПЗ до магістерської кваліфікаційної роботи;

– програмний додаток;

– презентація;

– відзив керівника роботи;

– відзив опонента

Технічне завдання до виконання прийняв _____ Дроганов Д.О.

Додаток Б. Скрипти створення бази даних

```
USE [master]
GO
/****** Object:  Database [DB]    Script Date: 13.11.2023 13:36:25 ******/
CREATE DATABASE [DB]
 CONTAINMENT = NONE
 ON  PRIMARY
( NAME = N'DB', FILENAME = N'C:\Program Files (x86)\Microsoft SQL
Server\MSSQL12.SQLEXPRESS\MSSQL\DATA\DB.mdf' , SIZE = 5120KB , MAXSIZE =
UNLIMITED, FILEGROWTH = 1024KB )
 LOG ON
( NAME = N'DB_log', FILENAME = N'C:\Program Files (x86)\Microsoft SQL
Server\MSSQL12.SQLEXPRESS\MSSQL\DATA\DB_log.ldf' , SIZE = 1024KB , MAXSIZE =
2048GB , FILEGROWTH = 10%)
GO
ALTER DATABASE [DB] SET DELAYED_DURABILITY = DISABLED
GO
USE [DB]
GO
/****** Object:  Table [dbo].[Categories]    Script Date: 13.11.2023 13:36:25 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Categories](
	[CategoriesId] [int] IDENTITY(1,1) NOT NULL,
	[CategoriesName] [nvarchar](220) NULL,
	[Description] [nvarchar](max) NULL,
PRIMARY KEY CLUSTERED
(
	[CategoriesId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
/****** Object:  Table [dbo].[CoapDatas]    Script Date: 13.11.2023 13:36:25 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[CoapDatas](
	[CoapDatasId] [int] IDENTITY(1,1) NOT NULL,
	[Mid] [float] NULL,
	[Code] [float] NULL,
	[OptDesc] [float] NULL,
	[LocationQuery] [float] NULL,
	[MaxAge] [float] NULL,
	[UriHost] [float] NULL,
	[ResponseIn] [float] NULL,
	[Retransmitted] [float] NULL,
```

```
	[Token] [float] NULL,
	[TokenLen] [float] NULL,
	[Class] [bit] NULL,
	[CategoriesId] [int] NULL,
	[IsGuessed] [bit] NULL,
PRIMARY KEY CLUSTERED
(
	[CoapDatasId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/****** Object:  Table [dbo].[Logs]    Script Date: 13.11.2023 13:36:25 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Logs](
	[LogsId] [int] IDENTITY(1,1) NOT NULL,
	[UsersId] [int] NULL,
	[EventNameShow] [nvarchar](max) NULL,
	[EventDate] [datetime] NULL,
	[UsersName] [nvarchar](100) NULL,
PRIMARY KEY CLUSTERED
(
	[LogsId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
/****** Object:  Table [dbo].[NeuralNetwork]    Script Date: 13.11.2023 13:36:25 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[NeuralNetwork](
	[NeuralNetworkId] [int] IDENTITY(1,1) NOT NULL,
	[NeuralNetworkNames] [nvarchar](200) NULL,
	[CategoriesId] [int] NULL,
	[NeuralNetworkFileModel] [nvarchar](max) NULL,
PRIMARY KEY CLUSTERED
(
	[NeuralNetworkId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
/****** Object:  Table [dbo].[SmartHomeSensorData]    Script Date: 13.11.2023 13:36:25 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
```

```
CREATE TABLE [dbo].[SmartHomeSensorData](
	[SmartHomeSensorDataId] [int] IDENTITY(1,1) NOT NULL,
	[Timestamp] [datetime] NULL,
	[IsMotionDetected] [bit] NULL,
	[Temperature] [float] NULL,
	[IsDoorOpen] [bit] NULL,
	[IsWindowOpen] [bit] NULL,
	[LightLevel] [float] NULL,
	[IsThereNoise] [bit] NULL,
	[Humidity] [float] NULL,
	[CameraFeedPath] [nvarchar](max) NULL,
	[AirQualityIndex] [float] NULL,
	[Room] [nvarchar](100) NULL,
PRIMARY KEY CLUSTERED
(
	[SmartHomeSensorDataId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
/****** Object:  Table [dbo].[Users]    Script Date: 13.11.2023 13:36:25 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Users](
	[UsersId] [int] IDENTITY(1,1) NOT NULL,
	[FirstName] [nvarchar](50) NULL,
	[LastName] [nvarchar](50) NULL,
	[UsersName] [nvarchar](50) NULL,
	[UsersPassword] [nvarchar](max) NULL,
	[RoleId] [int] NULL,
	[Description] [nvarchar](max) NULL,
	[Email] [nvarchar](max) NULL,
PRIMARY KEY CLUSTERED
(
	[UsersId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
USE [master]
GO
ALTER DATABASE [DB] SET  READ_WRITE
GO
```

Додаток В. Код для керування контролером системи моніторингу

```
#include <DHT.h>
#include <SoftwareSerial.h>

// Піни та налаштування для датчиків
#define DHTPIN 2
#define DHTTYPE DHT22
#define PIR_PIN 3
#define REED_PIN 4
#define LDR_PIN A0
#define MIC_PIN A1

DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  dht.begin();
  pinMode(PIR_PIN, INPUT);
  pinMode(REED_PIN, INPUT);
  pinMode(LDR_PIN, INPUT);
  pinMode(MIC_PIN, INPUT);
}

void loop() {
  // Читання даних з датчиків
  bool isMotion = digitalRead(PIR_PIN);
  bool isDoorOpen = digitalRead(REED_PIN);
  int lightLevel = analogRead(LDR_PIN);
  int soundLevel = analogRead(MIC_PIN);
  float temp = dht.readTemperature();
  float hum = dht.readHumidity();

  // Формування рядка з даними
  String dataString = String(isMotion) + «,» + String(temp) + «,» + String(isDoorOpen) + «,» +
String(lightLevel) + «,» + String(soundLevel) + «,» + String(hum);

  // Відправка даних на веб-сервер
  Serial.println(dataString);

  delay(2000); // Затримка між зчитуваннями
}
```

Додаток Г. Лістинги програми

Лістинг 1. Код класу «PredictTestModelForm»

```
using GuardProtectionApp.Providers;
using Microsoft.ML;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Runtime.Remoting.Metadata.W3cXsd2001;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace GuardProtectionApp.Forms.Controls {
  public partial class PredictTestModelForm : Form {
    private NeuralNetwork _SelectedNeural = new NeuralNetwork();
    private MLContext context = new MLContext();
    private PredictionEngine<SmartHomeData, SmartHomePrediction> predictor;
    private NeuralNetworkProvider _NeuralNetworkProvider = new NeuralNetworkProvider();

    private CategoriesProvider _CategoriesProvider = new CategoriesProvider();
    private List<Categories> _CategoriesList = new List<Categories>();
    private bool _IsThemesLoad = false;

    private SmartHomeDataGenerator _SmartHomeDataGenerator = new SmartHomeDataGenerator();

    public PredictTestModelForm() {
      InitializeComponent();
      LoadAllDate();
    }

    private void LoadAllDate() {
      _CategoriesList = _CategoriesProvider.GetAllCategories();
      CategoriesCBox.DataSource = _CategoriesList;
      CategoriesCBox.ValueMember = «CategoriesId»;
      CategoriesCBox.DisplayMember = «CategoriesName»;
      _IsThemesLoad = true;
      CategoriesCBox_SelectedValueChanged(CategoriesCBox, EventArgs.Empty);
    }

    private void LoadData(string FilePath) {
      string localProj = Application.StartupPath + FilePath;
      // Define DataViewSchema for data preparation pipeline and trained model
      DataViewSchema modelSchema;

      // Load trained model
      ITransformer model = context.Model.Load(localProj, out modelSchema);
```

```
    // Evaluate the model
    // Use the model to make predictions
    predictor = context.Model.CreatePredictionEngine<SmartHomeData,
SmartHomePrediction>(model);
  }

  private void RunBtn_Click(object sender, EventArgs e) {
    if (timer1.Enabled) {
      timer1.Enabled = false;
      RunBtn.Text = «Запустити»;
    } else {
      timer1.Enabled = true;
      RunBtn.Text = «Зупинити»;
    }
  }

  private void CategoriesCBox_SelectedValueChanged(object sender, EventArgs e) {
    if (_IsThemesLoad) {
      _SelectedNeural = _NeuralNetworkProvider.SelectedNeuralNetworkByCategoriesId(
        Convert.ToInt32(CategoriesCBox.SelectedValue));
      LoadData(_SelectedNeural.NeuralNetworkFileModel);
    }
  }

  private void timer1_Tick(object sender, EventArgs e) {
    var randomData = _SmartHomeDataGenerator.GenerateRandomData();

    // Додайте згенеровані дані до RaportTBox
    RaportTBox.Text = $»Згенеровані дані:\r\n Час початку: {randomData.StartTime}\r\n « +
                $»Місцезнаходження: {randomData.Location}\r\n « +
                $»Час доби: {randomData.TimeOfDay}\r\n « +
                $»Об'єкт: {randomData.Object}\r\n « +
                $»Поза: {randomData.Posture}\r\n « +
                $»Тривалість: {randomData.Duration}\r\n « +
                $»Діяльність: {randomData.Activity}\r\n»;

    // Прогнозування за допомогою моделі
    var prediction = predictor.Predict(randomData);
    bool isPotentialIntrusion = prediction.IsPotentialIntrusion();

    // Виведення результатів прогнозування
    RaportTBox.Text += $»Передбачувана діяльність: {prediction.PredictedActivity}\r\n»;
    RaportTBox.Text += isPotentialIntrusion ? «Можливе вторгнення виявлено!» : «Вторгнення не
виявлено.» + «\r\n»;

    // Прокрутити текстове поле до останнього запису
    RaportTBox.SelectionStart = RaportTBox.Text.Length;
    RaportTBox.ScrollToCaret();
  }


}
```

```
}

class SmartHomeDataGenerator {
  private static Random random = new Random();
  private static List<string> locations = new List<string> { «Bedroom», «Bathroom», «Kitchen», «Toilet», «Front» };
  private static List<string> timesOfDay = new List<string> { «Morning», «Afternoon», «Evening», «Night» };
  private static List<string> objects = new List<string> { «Hall-Bedroom door», «Hall-Bathroom door», «Plates cupboard», «Hall-Toilet door», «Frontdoor» };
  private static List<string> postures = new List<string> { «Sitting», «Standing» };
  private static List<string> activities = new List<string> { «go to bed», «use toilet», «prepare Breakfast», «take shower», «leave house», «door forced», «window broken», «unauthorized entry» };
  private static List<string> intrusionActivities = new List<string> { «door forced», «window broken», «unauthorized entry» };
  private static List<string> intrusionTimesOfDay = new List<string> { «Night» };

  public SmartHomeData GenerateRandomData() {
    bool isIntrusion = random.NextDouble() < 0.2; // 20% chance to generate intrusion data
    string generatedTime = GenerateRandomTime(isIntrusion);
    return new SmartHomeData {
      StartTime = generatedTime,
      Location = locations[random.Next(locations.Count)],
      TimeOfDay = DetermineTimeOfDay(generatedTime),
      Object = objects[random.Next(objects.Count)],
      Posture = postures[random.Next(postures.Count)],
      Duration = random.Next(30, 600),
      Activity = isIntrusion ? intrusionActivities[random.Next(intrusionActivities.Count)] : activities[random.Next(activities.Count)]
    };
  }

  private string GenerateRandomTime(bool isIntrusion) {
    int hour = isIntrusion ? random.Next(0, 4) : random.Next(0, 24);
    int minute = random.Next(0, 60);
    int second = random.Next(0, 60);
    return $»{hour:D2}:{minute:D2}:{second:D2}»;
  }

  private string DetermineTimeOfDay(string generatedTime) {
    var timeParts = generatedTime.Split(':');
    int hour = int.Parse(timeParts[0]);

    // Adjust hour range for 24-hour format
    if (hour < 0 || hour > 23) hour = 0; // default to 0 in case of invalid hour

    if (hour >= 6 && hour < 12) return «Morning»;
    if (hour >= 12 && hour < 17) return «Afternoon»;
    if (hour >= 17 && hour < 21) return «Evening»;
    return «Night»;
  }
```

}

Лістинг 2. Код класу «TrainForm»

```
using Microsoft.ML;
using Microsoft.ML.Data;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using GuardProtectionApp.AppCode;
using GuardProtectionApp.Forms.Systems;
using GuardProtectionApp.Providers;
using Microsoft.ML.Calibrators;
using Microsoft.ML.Trainers;

namespace GuardProtectionApp.Forms.Dictionary {
  public partial class TrainForm : Form {
    private string _Path = ««;
    private MLContext context;
    private ITransformer model;
    private IDataView data;

    private int _selectedRowIndex = 0;
    private CategoriesProvider _CategoriesProvider = new CategoriesProvider();
    private List<Categories> _CategoriesList = new List<Categories>();
    private ValidationMy _Validation = new ValidationMy();
    private NeuralNetworkProvider _NeuralNetworkProvider = new NeuralNetworkProvider();
    private List<NeuralNetwork> _NeuralNetworkList = new List<NeuralNetwork>();
    private LogsProvider _LogsProvider = new LogsProvider();
```

```
private bool _IsModelTrain = false;


public TrainForm() {
 InitializeComponent();
 LoadAllDate();
 DataLoad();
}

private void LoadAllDate() {
 _CategoriesList = _CategoriesProvider.GetAllCategories();
 CategoriesCBox.DataSource = _CategoriesList;
 CategoriesCBox.ValueMember = «CategoriesId»;
 CategoriesCBox.DisplayMember = «CategoriesName»;
}

private void DataLoad() {
 int firstRowIndex = 0;
 if (NeuralNetworkGridView.FirstDisplayedScrollingRowIndex > 0) {
  firstRowIndex = NeuralNetworkGridView.FirstDisplayedScrollingRowIndex;
 }
 try {
  _NeuralNetworkList = _NeuralNetworkProvider.GetAllNeuralNetwork();
  LoadDataInNeuralNetworkGridView(_NeuralNetworkList);
  if (_selectedRowIndex == NeuralNetworkGridView.Rows.Count) {
   _selectedRowIndex = NeuralNetworkGridView.Rows.Count - 1;
  }
  if (_selectedRowIndex >= 0) {
   NeuralNetworkGridView.FirstDisplayedScrollingRowIndex = firstRowIndex;
   NeuralNetworkGridView.Rows[_selectedRowIndex].Selected = true;
  }
 } catch (Exception ex) {
  MessageBox.Show(ex.ToString());
 }
}
```

```
        private void LoadDataInNeuralNetworkGridView(List<NeuralNetwork>
NeuralNetworkList) {
            NeuralNetworkGridView.DataSource = null;
            NeuralNetworkGridView.Columns.Clear();
            NeuralNetworkGridView.AutoGenerateColumns = false;
            NeuralNetworkGridView.RowHeadersVisible = false;

            NeuralNetworkGridView.DataSource = NeuralNetworkList;

            if (NeuralNetworkList.Count > 0) {
              if (NeuralNetworkList[0].Message ==
NamesMy.NoDataNames.NoDataInNeuralNetwork) {
                DataGridViewColumn messageColumn = new DataGridViewTextBoxColumn();
                messageColumn.DataPropertyName = «Message»;
                messageColumn.Width = NeuralNetworkGridView.Width -
NamesMy.SizeOptins.MinusSizePanel;
                NeuralNetworkGridView.Columns.Add(messageColumn);
              } else {
                DataGridViewColumn DetailIdColumn = new DataGridViewTextBoxColumn();
                DetailIdColumn.DataPropertyName = «NeuralNetworkId»;
                NeuralNetworkGridView.Columns.Add(DetailIdColumn);
                NeuralNetworkGridView.Columns[0].Visible = false;

                DataGridViewColumn numberColumn = new DataGridViewTextBoxColumn();
                numberColumn.HeaderText = «№ «;
                numberColumn.DataPropertyName = «Number»;
                numberColumn.DefaultCellStyle.Alignment =
DataGridViewContentAlignment.MiddleRight;
                numberColumn.Width = NamesMy.SizeOptins.NumberSize;
                NeuralNetworkGridView.Columns.Add(numberColumn);

                DataGridViewColumn NeuralNetworkNamesColumn = new
DataGridViewTextBoxColumn();
```

```
            NeuralNetworkNamesColumn.HeaderText = «Назва мережі»;
            NeuralNetworkNamesColumn.DataPropertyName = «NeuralNetworkNames»;
            NeuralNetworkNamesColumn.Width = 300;
            NeuralNetworkGridView.Columns.Add(NeuralNetworkNamesColumn);

            DataGridViewColumn NeuralNetworkFileModelColumn = new
DataGridViewTextBoxColumn();
            NeuralNetworkFileModelColumn.HeaderText = «Файл»;
            NeuralNetworkFileModelColumn.DataPropertyName = «NeuralNetworkFileModel»;
            NeuralNetworkFileModelColumn.Width = 300;
            NeuralNetworkGridView.Columns.Add(NeuralNetworkFileModelColumn);

            DataGridViewButtonColumn IsResidesBtn = new DataGridViewButtonColumn();
            IsResidesBtn.HeaderText = «Видалити»;
            IsResidesBtn.Text = «Видалити»;
            IsResidesBtn.UseColumnTextForButtonValue = true;
            IsResidesBtn.ToolTipText = «Видалити»;
            IsResidesBtn.Width = NamesMy.SizeOptins.DeleteBtnSize;
            NeuralNetworkGridView.Columns.Add(IsResidesBtn);

          }
          for (int i = 0; i < NeuralNetworkGridView.Columns.Count; i++) {
            NeuralNetworkGridView.Columns[i].HeaderCell.Style.BackColor = Color.LightGray;
          }
        }
      }


      private void OpenBtn_Click(object sender, EventArgs e) {
        OpenFileDialog openFileDialog = new OpenFileDialog();
        openFileDialog.InitialDirectory = «C:\\»;
        openFileDialog.Filter = «Text files (*.csv)|*.csv|All files (*.*)|*.*»;
        openFileDialog.FilterIndex = 2;
        openFileDialog.RestoreDirectory = true;
```

```
            if (openFileDialog.ShowDialog() == DialogResult.OK) {
              try {
                _Path = openFileDialog.FileName;
                FileNameTBox.Text = openFileDialog.FileName;

                context = new MLContext(seed: 0);
                data = context.Data.LoadFromTextFile<SmartHomeData>(
                   path: FileNameTBox.Text,
                   separatorChar: ',',
                   hasHeader: true);

                var trainTestSplit = context.Data.TrainTestSplit(data);
                var trainData = trainTestSplit.TrainSet;
                var testData = trainTestSplit.TestSet;

                var pipeline = context.Transforms.Conversion.MapValueToKey(«LabelKey», «Label»)
                  .Append(context.Transforms.Text.FeaturizeText(«LocationFeaturized», «Location»))
                  .Append(context.Transforms.Text.FeaturizeText(«TimeOfDayFeaturized»,
«TimeOfDay»))
                  .Append(context.Transforms.Text.FeaturizeText(«ObjectFeaturized», «Object»))
                  .Append(context.Transforms.Text.FeaturizeText(«PostureFeaturized», «Posture»))
                  .Append(context.Transforms.Concatenate(«Features», «LocationFeaturized»,
«TimeOfDayFeaturized», «ObjectFeaturized», «PostureFeaturized», «Duration»))
                  .Append(context.Transforms.CopyColumns(«Label», «LabelKey»));

                var trainer = context.MulticlassClassification.Trainers.LbfgsMaximumEntropy(«Label»,
«Features»);
                var trainingPipeline = pipeline.Append(trainer)
                   .Append(context.Transforms.Conversion.MapKeyToValue(«PredictedLabel»,
«Label»));

                RaportTBox.Text = «Training the model...\r\n»;
                model = trainingPipeline.Fit(trainData);
                var predictions = model.Transform(testData);
```

```
            var metrics = context.MulticlassClassification.Evaluate(predictions, «LabelKey»,
«Score»);
            RaportTBox.Text += $»Log-loss: {metrics.LogLoss - 0.5}\r\n»;
            RaportTBox.Text += $»Macro accuracy: {metrics.MacroAccuracy}\r\n»;
            RaportTBox.Text += $»Micro accuracy: {metrics.MicroAccuracy}\r\n»;
            RaportTBox.Text += $»Log-loss reduction: {metrics.LogLossReduction}\r\n»;

            _IsModelTrain = true;
          } catch (Exception ex) {
            MessageBox.Show($»Помилка: {ex.Message}»);
          }
        }
      }

      private void AddBtn_Click(object sender, EventArgs e) {
        if (IsDataEnteringCorrect()) {
          //Save model
          string pathName = @»\teach\» + GenerateFileName() + «.zip»;
          string localProj = System.IO.Path.GetDirectoryName(
            System.Reflection.Assembly.GetExecutingAssembly().Location);
          _NeuralNetworkProvider.InsertNeuralNetwork(NeuralNetworkNamesTBox.Text,
            Convert.ToInt32(CategoriesCBox.SelectedValue),
            pathName);
          context.Model.Save(model, data.Schema, localProj + pathName);
          //context.Model.Save(model, data.Schema, «model.zip»);
          ClearAllData();
          _LogsProvider.InsertLogs(LoginForm.CurrentUser.UsersId,
            «Було навчено нейронну мережу « +
            NeuralNetworkNamesTBox.Text, DateTime.Now);
          MessageBox.Show(«Дані успішно збережено!»);
        }
      }

      private void ClearBtn_Click(object sender, EventArgs e) {
        ClearAllData();
```

```
        }

        private void ExitBtn_Click(object sender, EventArgs e) {
          this.Close();
        }

        public string GenerateFileName() {
          DateTime now = DateTime.Now;
          string fileName = string.Format(«{0}_{1}_{2}_{3}_{4}_{5}»,
              now.Year, now.Month, now.Day, now.Hour, now.Minute, now.Second);

          return fileName;
        }

        private void ClearAllData() {
          _IsModelTrain = false;
          FileNameTBox.Text = String.Empty;
          NeuralNetworkNamesTBox.Text = String.Empty;
          RaportTBox.Text = String.Empty;
          DataLoad();
        }
        private bool IsDataEnteringCorrect() {
          bool isCorrect = true;
          if (!_IsModelTrain) {
            MessageBox.Show(«Неможливо зберегти дані. \r\nЩе не навчено нейронну
мережу!», «Увага!»);
            isCorrect = false;
          }
          if (Convert.ToInt32(CategoriesCBox.SelectedValue) > 0) {
            CategoriesValidationLbl.Text = NamesMy.ProgramButtons.RequiredValidation;
          } else {
            CategoriesValidationLbl.Text = NamesMy.ProgramButtons.ErrorValidation;
            isCorrect = false;
          }
          if (_Validation.IsDataEntering(NeuralNetworkNamesTBox.Text)) {
```

```
            NeuralNetworkNamesValidationLbl.Text =
NamesMy.ProgramButtons.RequiredValidation;
          } else {
            NeuralNetworkNamesValidationLbl.Text = NamesMy.ProgramButtons.ErrorValidation;
            isCorrect = false;
          }
          return isCorrect;
        }

        private void NeuralNetworkGridView_CellClick(object sender,
DataGridViewCellEventArgs e) {
          if (e.ColumnIndex == 4 && NeuralNetworkGridView[0, e.RowIndex].Value.ToString() !=
_NeuralNetworkList[0].Message) {
            if (MessageBox.Show(«Ви дійсно хочете видалити цей елемент?», «Видалити»,
MessageBoxButtons.YesNo) == DialogResult.Yes) {

_NeuralNetworkProvider.DeleteNeuralNetworkByNeuralNetworkId(Convert.ToInt32(NeuralNetwork
GridView[0, e.RowIndex].Value.ToString()));
              DataLoad();
            }
          }
        }
      }

      public class SmartHomeData {
        [LoadColumn(0)] public string StartTime { get; set; }
        [LoadColumn(1)] public string Location { get; set; }
        [LoadColumn(2)] public string TimeOfDay { get; set; }
        [LoadColumn(3)] public string Object { get; set; }
        [LoadColumn(4)] public string Posture { get; set; }
        [LoadColumn(5)] public float Duration { get; set; }
        [ColumnName(«Label»), LoadColumn(6)] public string Activity { get; set; }
      }
```

```
public class SmartHomePrediction {
  [ColumnName(«PredictedLabel»)] public string PredictedActivity { get; set; }
  public float[] Score { get; set; }

  public bool IsPotentialIntrusion() {
    // Define intrusion detection logic here
    // For example: if the predicted activity is «unidentified entry» or if the time is unusual (like «3 AM»)
    var unusualActivities = new HashSet<string> { «unidentified entry», «door forced», «window broken», «unauthorized entry» };
    return unusualActivities.Contains(PredictedActivity);
  }
}
```

Лістинг 3. Код класу «SmartHomeSensorDataProvider»

```
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Data;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace GuardProtectionApp.Providers {
  internal class SmartHomeSensorDataProvider {
    private string _ConnString = System.Configuration.ConfigurationSettings.AppSettings[«CONNECT»];

    // Метод для вставки даних сенсорів у базу даних
    public void InsertSmartHomeSensorData(SmartHomeSensorData data) {
      string SqlString = «INSERT INTO SmartHomeSensorData (Timestamp, IsMotionDetected, Temperature, IsDoorOpen, IsWindowOpen, LightLevel, IsThereNoise, Humidity, CameraFeedPath, AirQualityIndex, Room) Values(@Timestamp, @IsMotionDetected, @Temperature, @IsDoorOpen, @IsWindowOpen, @LightLevel, @IsThereNoise, @Humidity, @CameraFeedPath, @AirQualityIndex, @Room)»;

      using (SqlConnection conn = new SqlConnection(_ConnString)) {
        using (SqlCommand cmd = new SqlCommand(SqlString, conn)) {
          cmd.CommandType = CommandType.Text;
          cmd.Parameters.AddWithValue(«@Timestamp», data.Timestamp);
          cmd.Parameters.AddWithValue(«@IsMotionDetected», data.IsMotionDetected);
          cmd.Parameters.AddWithValue(«@Temperature», data.Temperature);
          cmd.Parameters.AddWithValue(«@IsDoorOpen», data.IsDoorOpen);
          cmd.Parameters.AddWithValue(«@IsWindowOpen», data.IsWindowOpen);
          cmd.Parameters.AddWithValue(«@LightLevel», data.LightLevel);
          cmd.Parameters.AddWithValue(«@IsThereNoise», data.IsThereNoise);
```

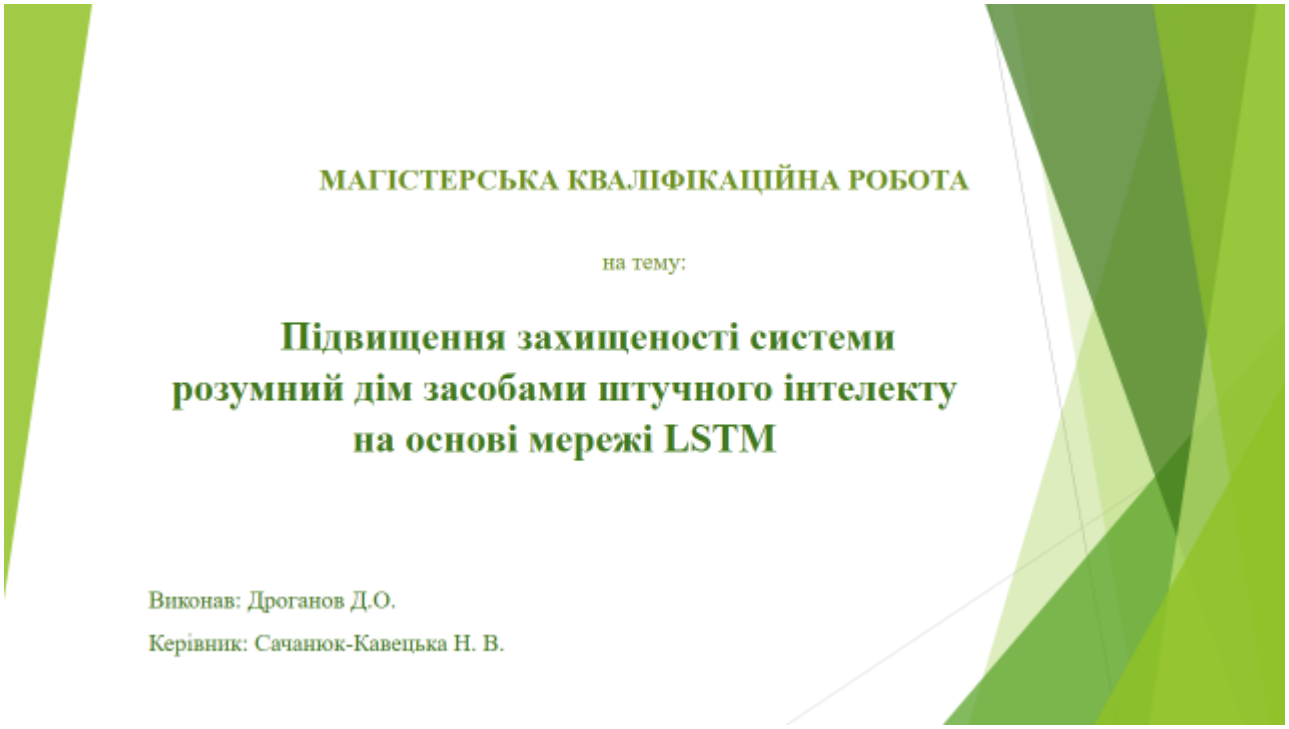
```
        cmd.Parameters.AddWithValue(«@Humidity», data.Humidity);
        cmd.Parameters.AddWithValue(«@CameraFeedPath», data.CameraFeedPath ??
(object)DBNull.Value); // Handle null values
        cmd.Parameters.AddWithValue(«@AirQualityIndex», data.AirQualityIndex);
        cmd.Parameters.AddWithValue(«@Room», data.Room);
        conn.Open();
        cmd.ExecuteNonQuery();
        conn.Close();
      }
    }
  }

  // Метод для отримання всіх даних сенсорів з бази даних
  public List<SmartHomeSensorData> GetAllSmartHomeSensorData() {
    string SqlString = «SELECT * FROM SmartHomeSensorData ORDER BY Timestamp DESC»; //
Чи інший критерій сортування

    List<SmartHomeSensorData> listAllSensorData = new List<SmartHomeSensorData>();
    using (SqlConnection conn = new SqlConnection(_ConnString)) {
      using (SqlCommand cmd = new SqlCommand(SqlString, conn)) {
        conn.Open();
        using (SqlDataReader reader = cmd.ExecuteReader()) {
          while (reader.Read()) {
            SmartHomeSensorData oneSensorData = new SmartHomeSensorData();
            // Припускаючи, що клас SmartHomeSensorData має відповідний конструктор
            oneSensorData.Timestamp = reader.GetDateTime(reader.GetOrdinal(«Timestamp»));
            oneSensorData.IsMotionDetected =
reader.GetBoolean(reader.GetOrdinal(«IsMotionDetected»));
            oneSensorData.Temperature = reader.GetDouble(reader.GetOrdinal(«Temperature»));
            oneSensorData.IsDoorOpen = reader.GetBoolean(reader.GetOrdinal(«IsDoorOpen»));
            oneSensorData.IsWindowOpen = reader.GetBoolean(reader.GetOrdinal(«IsWindowOpen»));
            oneSensorData.LightLevel = reader.GetDouble(reader.GetOrdinal(«LightLevel»));
            oneSensorData.IsThereNoise = reader.GetBoolean(reader.GetOrdinal(«IsThereNoise»));
            oneSensorData.Humidity = reader.GetDouble(reader.GetOrdinal(«Humidity»));
              oneSensorData.CameraFeedPath = reader.IsDBNull(reader.GetOrdinal(«CameraFeedPat;»))
? null : reader.GetString(reader.GetOrdinal(«CameraFeedPath»));
            oneSensorData.AirQualityIndex = reader.GetDouble(reader.GetOrdinal(«AirQualityIndex»));
              oneSensorData.Room = reader.GetString(reader.GetOrdinal(«Room»));
            listAllSensorData.Add(oneSensorData);
          }
        }
        conn.Close();
      }
    }

    return listAllSensorData;
  }

  // Метод для вибору конкретних даних сенсора за ідентифікатором
  public SmartHomeSensorData SelectSmartHomeSensorDataById(int id) {
    string SqlString = «SELECT * FROM SmartHomeSensorData WHERE Id=@Id»;
```

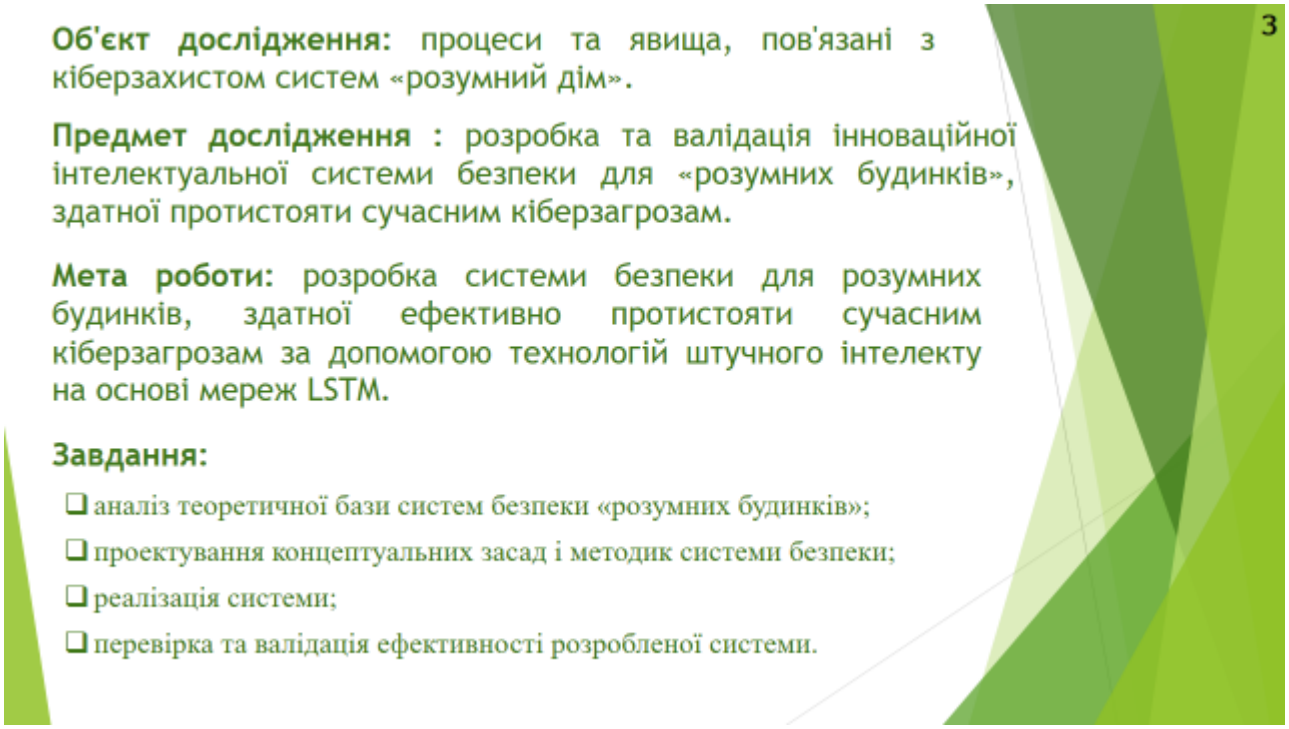
```
    SmartHomeSensorData sensorData = null;
    using (SqlConnection conn = new SqlConnection(_ConnString)) {
     using (SqlCommand cmd = new SqlCommand(SqlString, conn)) {
      cmd.Parameters.AddWithValue(«@Id», id);
      conn.Open();
      using (SqlDataReader reader = cmd.ExecuteReader()) {
       if (reader.Read()) {
        sensorData = new SmartHomeSensorData {
         // Припускається, що клас SmartHomeSensorData має відповідний конструктор або
публічні сеттери
         // Наповнення даними з reader...
        };
       }
      }
      conn.Close();
     }
    }
    return sensorData;
   }

   // Метод для оновлення даних сенсора
   public void UpdateSmartHomeSensorData(SmartHomeSensorData data) {
    string SqlString = @»
     UPDATE SmartHomeSensorData
     SET
        Timestamp = @Timestamp,
        IsMotionDetected = @IsMotionDetected,
        Temperature = @Temperature,
        IsDoorOpen = @IsDoorOpen,
        IsWindowOpen = @IsWindowOpen,
        LightLevel = @LightLevel,
        IsThereNoise = @IsThereNoise,
        Humidity = @Humidity,
        CameraFeedPath = @CameraFeedPath,
        AirQualityIndex = @AirQualityIndex,
        Room = @Room
     WHERE Id = @Id»;

    using (SqlConnection conn = new SqlConnection(_ConnString)) {
     using (SqlCommand cmd = new SqlCommand(SqlString, conn)) {
      cmd.CommandType = CommandType.Text;
      cmd.Parameters.AddWithValue(«@Timestamp», data.Timestamp);
      cmd.Parameters.AddWithValue(«@IsMotionDetected», data.IsMotionDetected);
      cmd.Parameters.AddWithValue(«@Temperature», data.Temperature);
      cmd.Parameters.AddWithValue(«@IsDoorOpen», data.IsDoorOpen);
      cmd.Parameters.AddWithValue(«@IsWindowOpen», data.IsWindowOpen);
      cmd.Parameters.AddWithValue(«@LightLevel», data.LightLevel);
      cmd.Parameters.AddWithValue(«@IsThereNoise», data.IsThereNoise);
      cmd.Parameters.AddWithValue(«@Humidity», data.Humidity);
      cmd.Parameters.AddWithValue(«@CameraFeedPath», (object)data.CameraFeedPath ??
DBNull.Value); // Use DBNull for null values
      cmd.Parameters.AddWithValue(«@AirQualityIndex», data.AirQualityIndex);
```

```
        cmd.Parameters.AddWithValue(«@Room», data.Room);
        cmd.Parameters.AddWithValue(«@SmartHomeSensorDataId», data.SmartHomeSensorDataId);

        conn.Open();
        cmd.ExecuteNonQuery();
        conn.Close();
      }
    }
  }


  // Метод для видалення даних сенсора за ідентифікатором
  public void DeleteSmartHomeSensorDataById(int SmartHomeSensorDataId) {
    string SqlString = «DELETE FROM SmartHomeSensorData WHERE
SmartHomeSensorDataId=@SmartHomeSensorDataId»;
    using (SqlConnection conn = new SqlConnection(_ConnString)) {
      using (SqlCommand cmd = new SqlCommand(SqlString, conn)) {
        cmd.Parameters.AddWithValue(«@SmartHomeSensorDataId», SmartHomeSensorDataId);
        conn.Open();
        cmd.ExecuteNonQuery();
        conn.Close();
      }
    }
  }

 }
}


public class SmartHomeSensorData {
 public int SmartHomeSensorDataId { get; set; }
 public DateTime Timestamp { get; set; }
 public bool IsMotionDetected { get; set; }
 public double Temperature { get; set; }
 public bool IsDoorOpen { get; set; }
 public bool IsWindowOpen { get; set; }
 public double LightLevel { get; set; }
 public bool IsThereNoise { get; set; }
 public double Humidity { get; set; }
 public string CameraFeedPath { get; set; }
 public double AirQualityIndex { get; set; }
 public string Room { get; set; }

 public SmartHomeSensorData() { }

 // Конструктор класу
 public SmartHomeSensorData(DateTime timestamp, bool isMotionDetected, double temperature,
                            bool isDoorOpen, bool isWindowOpen, double lightLevel,
                            bool isThereNoise, double humidity, string cameraFeedPath,
                            double airQualityIndex, string room) {
  Timestamp = timestamp;
  IsMotionDetected = isMotionDetected;
```

```
    Temperature = temperature;
    IsDoorOpen = isDoorOpen;
    IsWindowOpen = isWindowOpen;
    LightLevel = lightLevel;
    IsThereNoise = isThereNoise;
    Humidity = humidity;
    CameraFeedPath = cameraFeedPath;
    AirQualityIndex = airQualityIndex;
    Room = room;
  }

}
```

Додаток Д. Ілюстративний матеріал (презентація)





5

Концепція «Розумний будинок»

7

Основні компоненти інтелектуальних систем

9

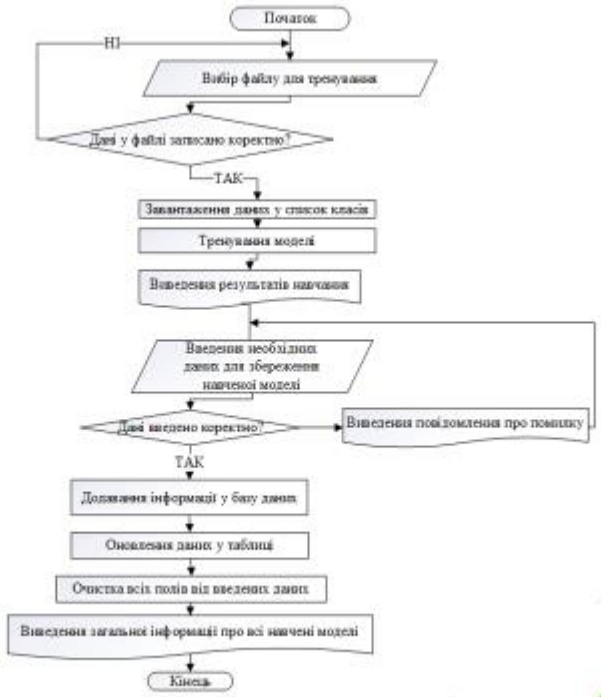
Алгоритм роботи мережі LSTM

1

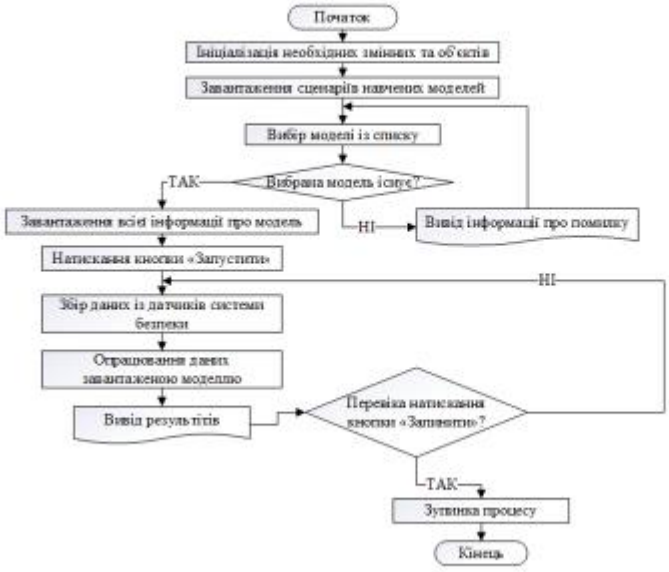
Вибір платформи розробки та мови програмування

- ❑ – мова програмування C#;
- ❑ – платформа .NET Framework 4.7;
- ❑ – середовище розробки MS Visual Studio 2022;
- ❑ – система управління базами даних MS SQL Server 2019;
- ❑ – інструмент автоматичного тестування коду MSTest;
- ❑ – бібліотека машинного навчання ML .NET.

Алгоритм навчання та зберігання даних моделі



Алгоритм роботи моделі



17

Діаграма класів рівня даних

19

Діаграма класів рівня користувацького інтерфейсу

21

Діаграма класів бізнес-логіки

23

Класи для навчання моделі

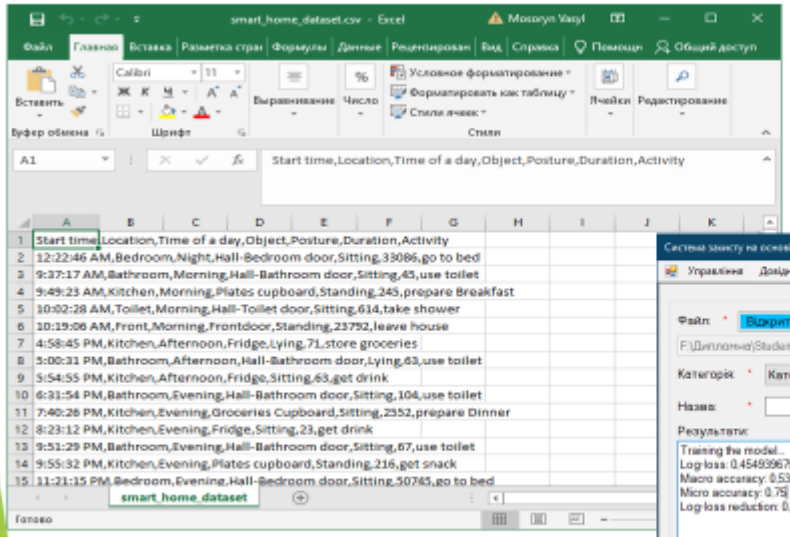
```
public class SmartHomeData {
    [LoadColumn(0)] public string StartTime { get; set; }
    [LoadColumn(1)] public string Location { get; set; }
    [LoadColumn(2)] public string TimeOfDay { get; set; }
    [LoadColumn(3)] public string Object { get; set; }
    [LoadColumn(4)] public string Posture { get; set; }
    [LoadColumn(5)] public float Duration { get; set; }
    [ColumnName("Label"), LoadColumn(6)] public string Activity { get; set; }
}
public class SmartHomePrediction {
    [ColumnName("PredictedLabel")] public string PredictedActivity { get; set; }
    public float[] Score { get; set; }
    public bool IsPotentialIntrusion() {
        var unusualActivities = new HashSet<string> { "unidentified entry", "door forced", "window broken", "unauthorized entry" };
        return unusualActivities.Contains(PredictedActivity);
    }
}
```

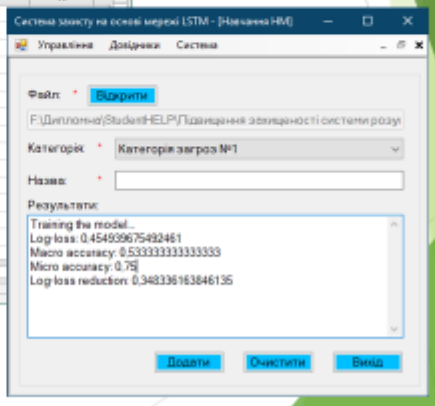
Код події «AddBtn_Click»

```
private void AddBtn_Click(object sender, EventArgs e) {
    if (IsDataEnteringCorrect()) {
        //Save model
        string pathName = @"\teach\" + GenerateFileName() + ".zip";
        string localProj = System.IO.Path.GetDirectoryName(
            System.Reflection.Assembly.GetExecutingAssembly().Location);
        _NeuralNetworkProvider.InsertNeuralNetwork(NeuralNetworkNamesTBox.Text,
            Convert.ToInt32(CategoriesCBox.SelectedValue),
            pathName);
        context.Model.Save(model, data.Schema, localProj + pathName);
        ClearAllData();
        _LogsProvider.InsertLogs(LoginForm.CurrentUser.UsersId,
            "Була навчена нейронну мережу " +
            NeuralNetworkNamesTBox.Text, DateTime.Now);
        MessageBox.Show("Дані успішно збережені!");
    }
}
```

25

Фрагмент вікна із історичними даними навчання

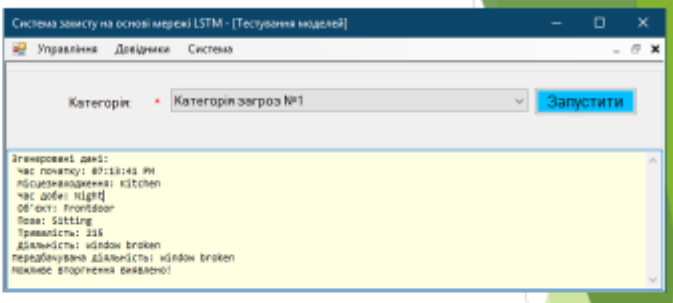




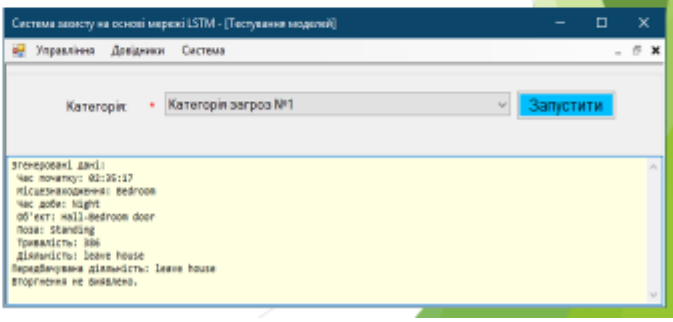
Форма результату навчання моделі

27

Проведення експериментів



Результат генерації 1-го сценарію



Результат генерації 2-го сценарію

ВИСНОВКИ

1. Проаналізовано концепцію розумних будинків.
2. Розроблено математичну модель на основі мереж LSTM для прогнозування поведінкових шаблонів у системах «Розумний дім».
3. Проведено аналіз потенційних загроз у системах «Розумний дім» та розроблено стратегії їх нейтралізації.
4. Експериментально підтверджено ефективність розробленої моделі LSTM у виявленні та реагуванні на потенційні загрози.
5. Визначено напрямки подальшого розвитку системи, включаючи інтеграцію з іншими компонентами та оптимізацію алгоритмів LSTM.
6. Проведено економічний аналіз проекту, що підтвердив його комерційну привабливість та потенціал для впровадження на ринку.

Дякую за увагу!

Додаток Е. Протокол перевірки на антиплагіат

ПРОТОКОЛ
ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Підвищення захищеності системи "розумний дім" засобами штучного інтелекту на основі мережі LSTM

Тип роботи: магістерська кваліфікаційна робота
(БДР, МКР)

Підрозділ: Кафедра менеджменту та безпеки інформаційних систем
Факультет менеджменту та інформаційної безпеки
(кафедра, факультет)

Показники звіту подібності Unicheck

Оригінальність 99 % Схожість 1 %

Аналіз звіту подібності (відмітити потрібне):
1. Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
2. Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
3. Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку _____ Коваль Н.П.
(підпис) (прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи _____ Дроганов Д.О.
(підпис) (прізвище, ініціали)

Керівник роботи _____ Сачанюк-Кавецька Н.В.
(підпис) (прізвище, ініціали)