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

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Вдосконалена система оцінювання знань на основі вдосконаленої моделі контролювання дій користувача з використанням технології відслідковування рухів та погляду користувача»

Виконала: ст. 2-го курсу, групи КІТС-23м
спеціальності 125– Кібербезпека та захист інформації
Освітня програма – Кібербезпека інформаційних технологій та систем
(шифр і назва напряму підготовки, спеціальності)

Гладка Вікторія
(прізвище та ініціали)

Керівник: д. ф., доц. каф. МБІС
Салієва О. В.
(прізвище та ініціали)
«____» _____ 2024 р.

Опонент: к.т.н., доцент, каф. ОТ
Богомолов С.В.
(прізвище та ініціали)
«____» _____ 2024 р.

Допущено до захисту
Голова секції УБ кафедри МБІС

_____ Юрій ЯРЕМЧУК
"09" грудня 2024 р.

Вінниця ВНТУ – 2024 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 та захист інформації
Освітньо-професійна програма – Кібербезпека інформаційних технологій та систем

ЗАТВЕРДЖУЮ
Голова секції УБ, кафедра МБІС

_____ **Юрій ЯРЕМЧУК**
"20" вересня 2024 р.

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

1. Тема роботи:

«Вдосконалена система оцінювання знань на основі вдосконаленої моделі контролювання дій користувача з використанням технології відслідковування рухів та погляду користувача»
Керівник роботи: д. ф., доц. каф. МБІС Салієва О.В.
(прізвище, ім`я, по-батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу від "17" вересня 2024 року № 310
2. Строк подання студентом роботи за тиждень до захисту.
3. Вихідні дані до роботи:
Стандарти, електронні джерела, підручники та наукові статті по темі, які стосуються теми магістерської кваліфікаційної роботи.
4. Зміст текстової частини:
Робота складається з трьох розділів. У першому розділі розглянуто теоретичні основи створення систем оцінювання знань, включаючи аналіз сучасних підходів до оцінювання, методів контролювання дій користувача, основ технологій відстежування рухів і погляду, а також проведено аналіз існуючих моделей контролювання. У другому розділі описано процес проєктування вдосконаленої системи, враховуючи особливості використання технологій моніторингу поведінки, розробку алгоритмів і моделі автоматизованого прийняття рішень. У третьому розділі представлено програмну реалізацію системи, обгрунтовано вибір технологій, реалізовано основні модулі системи, включаючи графічний інтерфейс, і проведено тестування для перевірки її ефективності

5. Перелік ілюстративного матеріалу (з точним зазначенням обов`язкових креслень)
У другому розділі магістерської кваліфікаційної роботи наведено 3 рисунки, у третьому розділі – 10 рисунків.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Основна частина			
I	Салієва О.В. д. ф., доц. каф. МБІС		
II	Салієва О.В. д. ф., доц. каф. МБІС		
III	Салієва О.В. д. ф., доц. каф. МБІС		
Економічна частина			
IV	Бурєннікова Н. В. д.е.н., професор каф. ЕПВМ		

7. Дата видачі завдання 20 вересня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи		Примітка
1.	Визначення напрямку магістерської роботи, формулювання теми	20.09.2024	31.09.2024	
2.	Аналіз предметної області обраної теми	01.10.2024	15.10.2024	
3.	Розробка роботи	16.10.2024	26.10.2024	
4.	Написання магістерської роботи на основі розробленої теми	27.10.2024	15.11.2024	
5.	Передзахист магістерської кваліфікаційної роботи	16.11.2024	24.11.2024	
6.	Виправлення, уточнення, корегування магістерської кваліфікаційної роботи	27.11.2024	04.12.2024	
7.	Захист магістерської кваліфікаційної роботи	10.12.2024	17.12.2024	

Студент _____ Гладка В.
(підпис)

Керівник роботи _____ Салієва О. В.
(підпис)

АНОТАЦІЯ

УДК 004.56.5

Гладка В. Магістерська кваліфікаційна робота зі спеціальності 125 – «Кібербезпека та захист інформації», освітня програма «Кібербезпека інформаційних технологій та систем». Вінниця: ВНТУ, 2024. – 134 с.

Укр. мовою. Бібліогр.: 41 назв; рис.: 32; табл. 9.

Ключові слова: відслідковування рухів, контроль дій, віслідковування погляду, безпека.

Дана магістерська робота присвячена розробці вдосконаленої системи оцінювання знань, яка базується на інтеграції моделі контролювання дій користувача із технологіями відстежування рухів і погляду. Актуальність роботи обумовлена необхідністю підвищення об’єктивності та чесності оцінювання знань у сучасних освітніх системах, особливо в умовах дистанційного навчання, де зростає ризик шахрайства.

У роботі проведено аналіз існуючих підходів до оцінювання знань, методів контролювання дій користувачів, а також технологій відстежування рухів та погляду. На основі отриманих результатів запропоновано нову модель, яка інтегрує поведінковий моніторинг з процесом тестування, що дозволяє оперативно виявляти підозрілу поведінку та автоматично приймати рішення про її вплив на результати тесту.

Програмна реалізація системи включає створення модулів для моніторингу рухів і погляду користувача, обробки результатів тестування та інтерактивного графічного інтерфейсу. Для цього використано сучасні технології, такі як OpenCV, Mediapipe, Flask, MongoDB, що забезпечило високу точність і ефективність роботи системи.

Результати роботи підтверджують, що впровадження таких систем дозволяє підвищити якість та об'єктивність освітніх процесів, знижуючи ризики шахрайства та забезпечуючи високий рівень довіри до результатів тестування.

ABSTRACT

UDC 004.56.5

Gladka V. Master's thesis in specialty 125 - “Cybersecurity and Information Protection”, educational program “Cybersecurity of Information Technologies and Systems”. Vinnytsia: VNTU, 2024. 134 p.

In Ukrainian. Bibliography: 41 titles; Figures: 32; Table 9.

Keywords: motion tracking, action control, gaze tracking, security.

This master's thesis is devoted to the development of an advanced knowledge assessment system based on the integration of a model for controlling user actions with motion and gaze tracking technologies. The relevance of the work is due to the need to increase the objectivity and fairness of knowledge assessment in modern educational systems, especially in distance learning, where the risk of fraud is increasing.

The paper analyzes existing approaches to knowledge assessment, methods of controlling user actions, as well as technologies for tracking movements and gaze. Based on the results obtained, a new model is proposed that integrates behavioral monitoring with the testing process, which allows to quickly detect suspicious behavior and automatically make decisions about its impact on the test results.

The software implementation of the system includes the creation of modules for monitoring user movements and gaze, processing test results, and an interactive graphical interface. For this purpose, modern technologies such as OpenCV, Mediapipe, Flask, MongoDB were used, which ensured high accuracy and efficiency of the system.

The results of the work confirm that the implementation of such systems can improve the quality and objectivity of educational processes, reducing the risks of fraud and ensuring a high level of confidence in the test results.

ЗМІСТ

ВСТУП

Актуальність. Сучасний розвиток цифрових технологій суттєво змінює підхід до освітнього процесу, зокрема до оцінювання знань. Автоматизовані системи тестування стають невід'ємною частиною навчання, особливо в умовах дистанційного навчання. Водночас зростає ризик шахрайства та нечесної поведінки під час тестування, що створює значні виклики для забезпечення об’єктивності та надійності оцінювання. Існуючі системи часто не враховують поведінкові аспекти користувачів, такі як рухи тіла чи погляд, які можуть бути індикаторами шахрайства. У зв'язку з цим актуальною є розробка вдосконаленої системи оцінювання знань, яка використовує інноваційні технології відстежування рухів та погляду користувача, що дозволить підвищити рівень контролю та забезпечити чесність у процесі тестування.

Мета і задачі дослідження. Метою дослідження є розробка вдосконаленої системи оцінювання знань, яка інтегрує технології моніторингу рухів і погляду користувача для підвищення точності оцінювання та зниження ризиків шахрайства під час тестування.

Для досягнення цієї мети було поставлено наступні задачі:

- провести аналіз існуючих систем оцінювання знань та методів контролю дій користувача.
- дослідити можливості використання технологій відстежування рухів та погляду в контексті моніторингу поведінки користувачів.
- розробити архітектуру вдосконаленої системи, що інтегрує модулі контролювання дій користувача з системою оцінювання знань.
- реалізувати програмні модулі для моніторингу рухів і погляду користувача, а також адаптивного тестування.
- забезпечити інтеграцію всіх компонентів системи та оцінити її ефективність у практичному застосуванні.

Об’єктом дослідження є процес оцінювання знань користувачів за допомогою автоматизованих тестових систем.

Предметом дослідження є методи і засоби вдосконалення систем оцінювання знань через інтеграцію технологій моніторингу поведінки користувачів.

Наукова новизна роботи полягає у розробці інноваційної моделі інтеграції систем відстежування рухів і погляду користувача з автоматизованими тестовими системами. Запропонована система забезпечує моніторинг поведінкових характеристик у реальному часі, що дозволяє оперативно виявляти підозрілу поведінку та значно знижувати рівень шахрайства під час тестування. Такий підхід не був широко реалізований у сучасних системах оцінювання знань.

Практична цінність роботи полягає у можливості впровадження розробленої системи у різноманітні освітні установи та платформи дистанційного навчання. Система дозволяє забезпечити високий рівень об'єктивності оцінювання знань, зменшити ймовірність нечесної поведінки та підвищити довіру до результатів тестування. Розроблені модулі можуть бути адаптовані для використання в інших сферах, наприклад, для моніторингу поведінки під час співбесід або сертифікаційних іспитів.

Апробація: тези доповідей представленні на Міжнародній науково-практичній Інтернет-конференції студентів, аспірантів та молодих науковців «Молодь в науці: дослідження, проблеми, перспективи (МН-2025)» [11].

1 ТЕОРЕТИЧНІ ЗАСАДИ СТВОРЕННЯ СИСТЕМИ ОЦІНЮВАННЯ ЗНАНЬ НА ОСНОВІ ВДОСКОНАЛЕНОЇ МОДЕЛІ КОНТРОЛЮВАННЯ ДІЙ КОРИСТУВАЧА

В даному розділі буде розглянуто теоретичні засади створення системи оцінювання знань, яка базується на вдосконаленій моделі контролювання дій користувача. Зокрема, увагу приділено сучасним підходам до оцінювання знань в освітніх системах, методам контролювання поведінки користувачів у комп’ютерних середовищах та основам технологій відслідковування рухів і погляду.

Також досліджено особливості застосування моделей контролювання для підвищення об’єктивності оцінювання знань, проведено аналіз існуючих моделей контролювання дій користувача та сформульовано основні задачі для подальшої розробки системи.

1.1 Сучасні підходи до оцінювання знань в освітніх системах

У сучасній освітній системі існує велике розмаїття підходів до оцінювання знань студентів та учнів. Метою оцінювання є визначення рівня знань, умінь, навичок, а також виявлення сильних і слабких сторін у навчальному процесі. Цей процес також відіграє роль зворотного зв'язку для коригування викладання і визначення рівня розуміння матеріалу студентами. Сучасні підходи відрізняються за типами використовуваних інструментів, критеріями, вимогами до об’єктивності та результативності, орієнтацією на компетентнісний підхід.

До традиційних методів належать письмові роботи, контрольні завдання, тести, усне опитування, реферати, та курсові проєкти. Вони є широко розповсюдженими завдяки простоті застосування та зрозумілості оцінювання. Основними характеристиками цих методів є їхня орієнтація на конкретний предметний зміст, можливість кількісного вимірювання результатів, але в багатьох випадках ці методи обмежують об'єктивність оцінювання через суб’єктивний підхід викладача [1].

Сучасні освітні системи переходять на компетентнісний підхід, що передбачає оцінювання не тільки знань, але й здатності до їх застосування в практичних ситуаціях. Компетентнісно орієнтовані методи включають портфоліо, самооцінку, оцінювання на основі проєктної діяльності, кейс-стаді та проблемно-орієнтовані завдання. Ці методи дозволяють комплексно оцінити знання та навички, але потребують додаткових ресурсів, як-от час на оцінювання та індивідуальний підхід до кожного студента.

Інформаційно-комунікаційні технології відкривають нові можливості для оцінювання, забезпечуючи точність та об’єктивність. Наприклад, онлайн-тестування з автоматичним підрахунком результатів, інтерактивні платформи з інтегрованими системами оцінювання, а також системи керування навчанням (LMS) дозволяють впроваджувати автоматизоване оцінювання знань у реальному часі. Це сприяє розвитку адаптивних навчальних середовищ, що дозволяють враховувати індивідуальні особливості учнів та їхній прогрес.

Інформаційно-комунікаційні технології кардинально змінили підходи до оцінювання знань, забезпечивши нові інструменти для вимірювання прогресу учнів та надання результатів в реальному часі. Інформаційно-комунікаційні технології дозволяють автоматизувати процеси тестування, аналітики та моніторингу, що значно спрощує роботу викладача і робить оцінювання об'єктивнішим. Завдяки цифровим платформам, викладачі отримали можливість використовувати інтерактивні засоби, адаптивне тестування та аналіз великих обсягів даних, що покращує точність та обґрунтованість оцінки [1].

Однією з найпоширеніших форм оцінювання, заснованих на інформаційно-комунікаційних технологіях, є онлайн-тестування. Цей метод дозволяє автоматизувати процес створення тестів, обробки відповідей та аналізу результатів. Інтерактивні платформи, такі як Google Classroom, Moodle, Canvas, Blackboard та інші, пропонують різні типи тестів (множинний вибір, короткі відповіді, завдання на відповідність тощо) і часто підтримують адаптивне налаштування тестів під рівень складності учня. Онлайн-тестування сприяє підвищенню об’єктивності оцінювання, оскільки воно мінімізує вплив людського фактору.

Штучний інтелект стає важливим елементом сучасних систем оцінювання. Використовуючи алгоритми машинного навчання, освітні платформи можуть підлаштовувати завдання та матеріали до індивідуальних особливостей учня. Адаптивне навчання дозволяє визначати рівень знань і компетенцій студента в реальному часі та пропонувати матеріали відповідної складності. Наприклад, якщо учень правильно відповідає на питання, система підвищує складність наступного питання, а у випадку помилки – знижує складність, що створює індивідуальний навчальний трек.

Аналітика навчальних даних, або Learning Analytics, дозволяє освітнім установам збирати і аналізувати великі обсяги даних про активність учнів у процесі навчання. Використовуючи Learning Analytics, викладачі можуть виявляти прогалини у знаннях, прогнозувати результати та адаптувати свої методи навчання та оцінювання для кращого розуміння потреб учнів. Наприклад, за допомогою аналітики можна відстежувати, як довго учень працює над завданнями, які питання викликають найбільші труднощі, і використовувати ці дані для подальшого персоналізованого підходу до навчання [2].

Інформаційно-комунікаційні технології дозволяють створювати інтерактивні завдання, що активізують інтерес учнів і забезпечують залучення до навчального процесу. Наприклад, завдання з використанням інтерактивних відеоматеріалів, віртуальних лабораторій, симуляцій та анімацій дозволяють учням краще засвоювати матеріал, що, у свою чергу, позитивно позначається на оцінюванні знань. Викладачі можуть інтегрувати в завдання аудіо- та відеоелементи, віртуальні тури та інші мультимедійні ресурси, що робить процес навчання і оцінювання цікавішим і ефективнішим.

У сучасних освітніх системах активно впроваджуються інноваційні підходи до оцінювання знань, орієнтовані на точну оцінку навичок та глибини розуміння учнів, а також на забезпечення чесності та прозорості тестування. Для запобігання списуванню та іншим формам шахрайства дедалі більше застосовуються технології контролю дій користувача, що дозволяють інтегрувати систему оцінювання знань

із технологіями відслідковування поведінки для підвищення її ефективності та достовірності результатів.

Штучний інтелект (ШІ) все більше застосовується для автоматизації оцінювання знань та аналізу результатів. ШІ може адаптувати складність завдань до рівня учня, аналізувати відповіді на відкриті питання, що дозволяє підвищити точність оцінювання. Окрім того, ШІ можна використовувати для виявлення патернів поведінки учнів, які можуть свідчити про списування, що дозволяє розробити більш комплексний підхід до забезпечення чесності під час тестування.

Технології відслідковування, як-от eye-tracking та motion-tracking, дозволяють виявляти ознаки несанкціонованої поведінки під час тестування. Наприклад, відстежування рухів очей дозволяє визначати, коли студент спрямовує погляд за межі екрану або надто часто відволікається. Ці технології стають ключовими елементами в системах контролю, орієнтованих на виявлення списування, особливо в системах, де немає прокторів [2].

Новітні підходи до оцінювання знань включають розробку програмних модулів, здатних виявляти несанкціоновані дії користувача і автоматично блокувати доступ до тесту або інтерфейсу системи. Наприклад, модуль може контролювати рухи учня, тривалість зосередження на завданні, частоту моргання та інші показники для виявлення підозрілої поведінки. Інтеграція таких модулів з відстежуванням рухів та погляду значно підвищує ефективність контролю та забезпечує чесність оцінювання.

Захист даних користувачів та забезпечення конфіденційності є важливими аспектами в системах оцінювання знань, які використовують відслідковування та інші технології моніторингу. Інформація про поведінку користувача, зокрема дані про рухи та погляд, є чутливою і вимагає надійного захисту від несанкціонованого доступу та витоку. Інтеграція методів шифрування, багатофакторної автентифікації та суворих правил доступу є необхідною умовою для збереження довіри учасників системи та дотримання етичних стандартів у навчанні [3].

1.2 Методи контролювання дій користувача в комп’ютерних системах

Контролювання дій користувача в комп'ютерних системах є критично важливим для забезпечення безпеки даних, запобігання шахрайству та підвищення точності оцінювання знань. Це стає особливо актуальним у контексті онлайн-оцінювання, де відсутність фізичного контролю робить традиційні методи перевірки недостатньо ефективними.

Відеоспостереження є одним з найпоширеніших методів контролю дій користувача в комп’ютерних системах, особливо під час тестування в онлайн-форматі. Цей метод передбачає використання камер для моніторингу поведінки учня, що дає змогу здійснювати фіксацію всіх дій у реальному часі.

Переваги:

– фіксація всіх дій користувача в реальному часі. Відеозапис дозволяє детально зафіксувати, як користувач взаємодіє з системою, включаючи його реакцію на запитання, жести, і навіть використання додаткових матеріалів. Можливість перегляду відео після завершення тестування дозволяє проводити ретельний аналіз;

– легкість у виявленні несанкціонованої активності. Відеоспостереження здатне швидко виявити, чи використовує учень сторонні матеріали, такі як підручники, шпаргалки чи інші пристрої. Використання камер забезпечує додатковий рівень контролю, оскільки знімає фізичні межі, які можуть бути порушені у звичайному тестуванні [4].

Недоліки:

– потребує значних ресурсів для зберігання та обробки відеоданих. Зберігання великих обсягів відеоінформації потребує значних витрат на апаратне забезпечення та програмне забезпечення, яке підтримуватиме цю функцію. Системи, що здійснюють відеоспостереження, повинні мати достатню пропускну здатність для передачі даних у реальному часі, що також потребує інвестицій;

– питання конфіденційності. Зйомка особи може сприйматися як вторгнення у приватне життя, що може викликати негативну реакцію користувачів. Важливо

враховувати етичні аспекти та отримувати згоду від користувачів на проведення відеоспостереження.

Моніторинг екрану передбачає запис дій, які виконуються на комп'ютері під час тестування. Це дозволяє виявляти, які програми відкриваються, які сайти відвідуються та які дії виконуються.

Переваги:

– відстеження доступу до Інтернету та використання сторонніх ресурсів. Моніторинг екрану дає змогу виявляти, чи використовує учень веб-сайти, що заборонені правилами тестування, або чи відкриває інші програми, які можуть бути використані для шахрайства;

– легкість у виявленні підозрілих дій. Запис екрану може швидко виявити, якщо користувач копіює текст з веб-сторінок або змінює вікна під час тестування, що може свідчити про спроби списування [5].

Недоліки:

– вимоги до великих обсягів пам'яті для зберігання записів. Збереження записів екрану потребує значних обсягів пам’яті, що може призвести до потреби в додаткових інвестиціях у зберігання даних;

– етичні питання щодо конфіденційності. Запис екрану може включати особисту інформацію, що підвищує ризик порушення приватності. Користувачі можуть відчувати дискомфорт через постійний моніторинг їхньої активності.

Системи програмного забезпечення для контролю активності користувачів забезпечують можливості моніторингу і можуть блокувати доступ до несанкціонованих ресурсів під час тестування.

Переваги:

– легкість у налаштуванні та використанні. Багато рішень для контролю мають інтуїтивно зрозумілий інтерфейс, що полегшує їхнє впровадження в систему;

– автоматичне блокування доступу. Програмне забезпечення може автоматично блокувати доступ до заборонених сайтів або додатків, що зменшує ризик використання сторонніх ресурсів.

Недоліки:

– обмежена ефективність. Користувачі можуть бути обізнані про наявність програм контролю і намагатися їх обійти, що зменшує загальну ефективність методів контролю;

– сумісність між операційними системами. Існує ризик, що програмне забезпечення не буде повністю сумісним з усіма версіями операційних систем, що може викликати проблеми під час тестування [6].

Технології, такі як eye-tracking (відслідковування погляду), motion-tracking (відслідковування рухів) і biometric tracking (біометричне відслідковування), забезпечують фіксацію рухів користувача в реальному часі.

Переваги:

– висока точність у виявленні несанкціонованих дій. Ці технології здатні виявляти не лише великі, але й найменші відхилення у поведінці користувача, що дозволяє оперативно реагувати на можливі випадки шахрайства;

– виявлення навіть найменших відхилень. Технології можуть зафіксувати такі деталі, як зміна погляду, що може свідчити про спробу обману.

Недоліки:

– необхідність спеціального обладнання. Для реалізації таких технологій потрібно додаткове обладнання, яке може бути дорогим та потребувати специфічного налаштування;

– питання етики та прав користувачів. Відслідковування фізичних рухів може бути сприйнято як вторгнення у приватність, що викликає етичні питання.

Застосування штучного інтелекту для аналізу поведінки користувачів може значно підвищити ефективність контролю [7].

Переваги:

– автоматичне виявлення підозрілих дій. Алгоритми машинного навчання можуть аналізувати дані в реальному часі і автоматично виявляти підозрілі дії, що дозволяє швидко реагувати на можливі спроби шахрайства;

– адаптація системи до нових форм шахрайства. Системи, що використовують ШІ, здатні вчитися на нових даних і підлаштовуватися під зміни в поведінці користувачів.

Недоліки:

– потребує великих обсягів даних для навчання. Для належної роботи алгоритмам необхідні великі обсяги даних, що може бути важко реалізувати;

– можливість помилок у розпізнаванні дій. Системи можуть неправильно інтерпретувати дії, що може призвести до хибних сповіщень і, відповідно, до неправильних рішень [8].

Біометрична аутентифікація використовує унікальні фізичні характеристики користувача для підтвердження його особи.

Переваги:

– високий рівень безпеки. Біометричні дані, такі як відбитки пальців або зображення обличчя, складно підробити, що підвищує рівень безпеки системи;

– швидкий доступ до системи. Користувачі можуть отримувати доступ до системи без необхідності запам'ятовувати паролі, що підвищує зручність використання.

Недоліки:

– високі витрати на впровадження технології. Впровадження біометричних систем потребує значних фінансових вкладень у технології та обладнання;

– зберігання біометричних даних вимагає суворих заходів безпеки, адже витік інформації може призвести до серйозних наслідків для приватності користувачів. Контроль дій користувачів у комп’ютерних системах порушує низку правових і етичних питань, що стають особливо актуальними в умовах розширення використання технологій моніторингу в освіті та інших сферах [9].

Законодавчі аспекти:

– у багатьох країнах існують закони, які регулюють обробку персональних даних. Наприклад, Загальний регламент захисту даних (GDPR) у ЄС вимагає явної згоди користувачів на збір даних. В Україні діє Закон України «Про захист персональних даних», який визначає порядок обробки особистої інформації та передбачає відповідальність за порушення цих норм;

– організації, що здійснюють моніторинг, несуть відповідальність за безпечне зберігання даних. У разі витоку даних можливі штрафи або інші санкції.

Права користувачів:

– користувачі мають право знати, які дані збираються, з якою метою та хто має доступ до них. Необхідно забезпечити механізми інформування і можливість відкликання згоди на обробку даних;

– організації зобов’язані захищати інформацію користувачів від несанкціонованого доступу, використовуючи шифрування, контроль доступу та регулярні аудити систем безпеки [9].

Етичні аспекти:

– використання технологій моніторингу може викликати відчуття втручання в приватне життя, що спричиняє дискомфорт у користувачів. Необхідно дотримуватися етичних норм, забезпечуючи прозорість процесів, рівні умови для всіх і уникнення дискримінації;

– постійний моніторинг може негативно впливати на психологічний стан користувачів, викликати стрес або дискомфорт, що знижує продуктивність.

Справедливість і транспарентність:

– системи контролю повинні бути налаштовані так, щоб уникнути упередженості або неправомірних звинувачень;

– важливо належно інформувати користувачів про мету та способи моніторингу, щоб зменшити недовіру і забезпечити відкритість процесу [10].

1.3 Аналіз технологій відслідковування рухів та погляду

Основою технологій відслідковування є можливість зчитування і обробки поведінкових патернів користувача, які відображають когнітивний стан та напрямок уваги. У контексті системи оцінювання знань ці технології можуть фіксувати, куди саме дивиться користувач, як часто відволікається та яким чином рухається його тіло під час виконання тесту. Принципи роботи технологій відслідковування базуються на таких етапах:

– збирання даних. Використання камер, інфрачервоних сенсорів або спеціальних датчиків для фіксації рухів очей, голови, тіла та навіть міміки користувача;

– обробка даних. Використання алгоритмів машинного навчання та комп’ютерного зору для аналізу рухів та виявлення аномалій;ф

– інтерпретація поведінкових патернів. Виявлення типових для списування моделей поведінки, які можуть включати постійне відведення погляду від екрану, часто повторювані рухи або специфічні пози [10].

Eye-tracking є однією з основних технологій відслідковування, яка базується на аналізі напрямку погляду користувача. У системах контролю за несанкціонованими діями eye-tracking дає можливість:

– визначати напрямок і фокусування погляду. Фіксація, куди саме користувач дивиться, може свідчити про його зосередженість на тесті або ж про спроби подивитися на інші джерела інформації;

– відстежувати зміну погляду та тривалість фіксацій. Занадто часта зміна напрямку погляду може свідчити про пошук підказок;

– аналіз моргання та рухів зіниць. Інтенсивність моргання або швидкість рухів зіниць можуть також давати уявлення про рівень концентрації та наявність відволікаючих факторів.

Eye-tracking може використовуватися для автоматичного блокування інтерфейсу або надсилання сповіщень адміністратору системи про підозрілу поведінку [11].

На рисунку 1.1 зображено схему роботи технології відслідковування погляду, розглянемо цю схему детальніше.

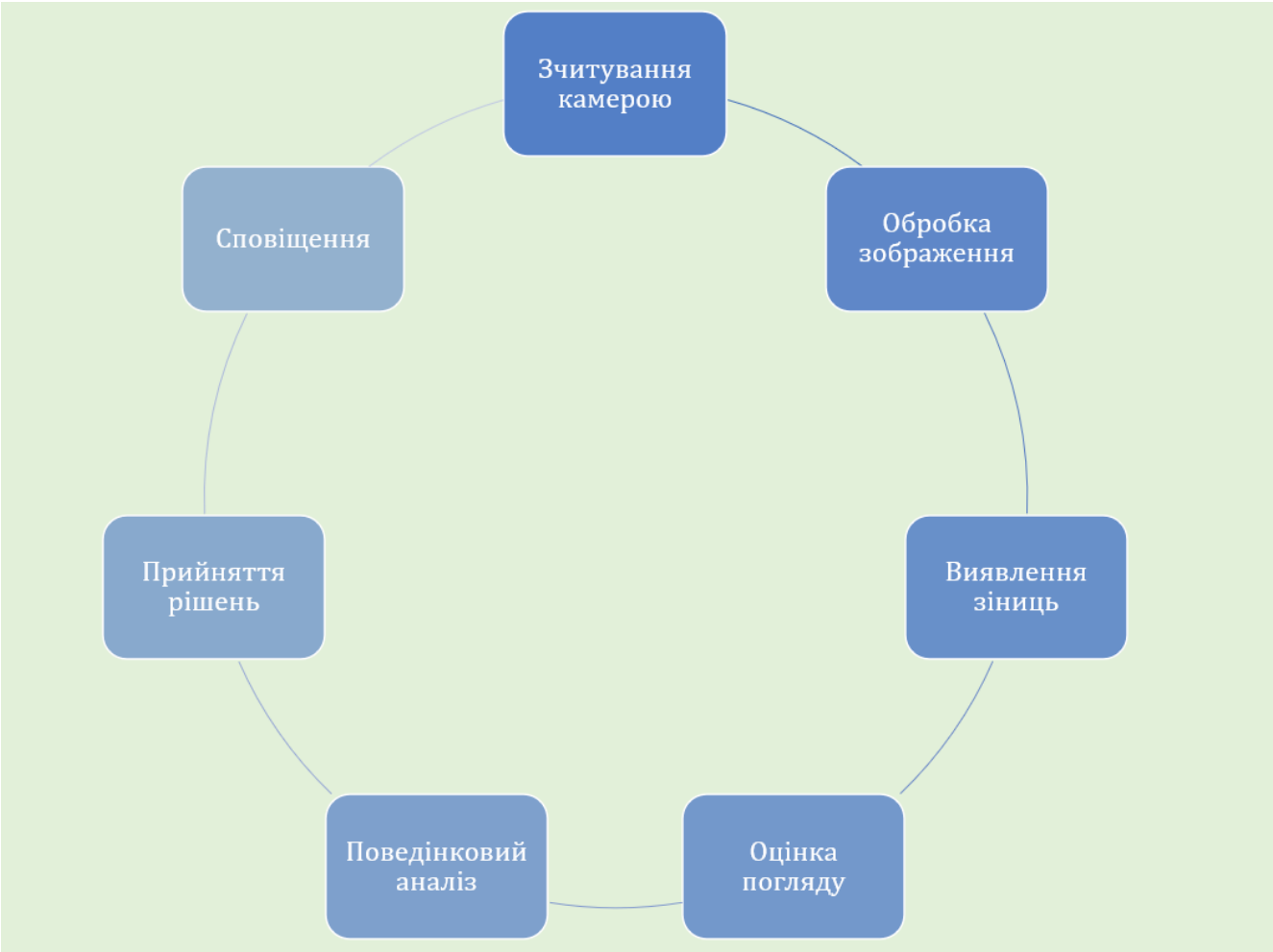


Рисунок 1.1 – Просто схема роботи технології відслідковування погляду

На першому етапі камера, інтегрована з системою, постійно зчитує зображення обличчя користувача. Камера може бути як звичайною, так і інфрачервоною, що підвищує точність роботи системи в умовах різного освітлення. Основна мета цього етапу — отримати якісні зображення обличчя для подальшого аналізу.

На другому етапі виконується первинна обробка зображення. Система відсіює зайві елементи, покращує якість зображення та виділяє області, де знаходяться очі користувача. Це спрощує подальше розпізнавання та робить аналіз більш ефективним [12].

Після обробки зображення система переходить до визначення точного положення зіниць. Використовуючи алгоритми комп'ютерного зору, система виявляє зіниці та інші маркери, щоб визначити, в якому напрямку дивиться користувач.

На цьому етапі система визначає, куди саме спрямований погляд користувача. Використовуючи координати зіниць і особливості положення очей, система розраховує напрямок погляду і фіксує зони екрану, на які дивиться користувач.

Система аналізує зібрані дані, щоб визначити загальний поведінковий патерн користувача під час виконання тесту. На основі напрямку та тривалості погляду, частоти моргань, а також пози тіла, система оцінює рівень зосередженості користувача. Якщо спостерігаються незвичайні чи підозрілі поведінкові патерни, система передає їх на подальший аналіз [13].

На шостому етапі система на основі проаналізованих даних приймає рішення щодо дій користувача. Якщо аналіз показує, що поведінка користувача має ознаки несанкціонованої дії, як-от спроби звертатися до сторонніх джерел або відволікання, система приймає рішення про наступні кроки.

Залежно від серйозності порушення, система може виконати певні дії, такі як:

– відправлення сповіщення адміністратору або проктору, який контролює процес тестування;

– блокування інтерфейсу користувача, що запобігає подальшому виконанню тесту;

– фіксація порушення в базі даних для аналізу і розробки рекомендацій з покращення безпеки в майбутньому.

Motion-tracking – це технологія, що дозволяє відстежувати загальну фізичну активність користувача, включаючи пози тіла, рухи голови, рук та інших частин тіла. Використання motion-tracking у системі контролю дій користувача під час тестування має такі переваги:

– контроль пози тіла. Визначення стандартної пози тестування та фіксація змін у ній. Наприклад, нахил голови може свідчити про спробу користувача звернутися до стороннього джерела;

– фіксація рухів, що свідчать про відволікання. Рухи тіла, що відображають неуважність або звернення до сторонніх ресурсів;

– виявлення типових патернів руху для кожного користувача. Система може адаптуватися до звичних рухів конкретного учня та швидше виявляти аномалії, що знижує ймовірність хибних спрацьовувань.

На рисунку 1.2 зображено основні етапи роботи системи motion-tracking для відстеження рухів користувача під час тестування. Ця система складається з декількох ключових етапів, які дозволяють фіксувати та аналізувати рухи користувача в режимі реального часу, а також приймати рішення про відповідні дії у разі виявлення підозрілої поведінки [14].

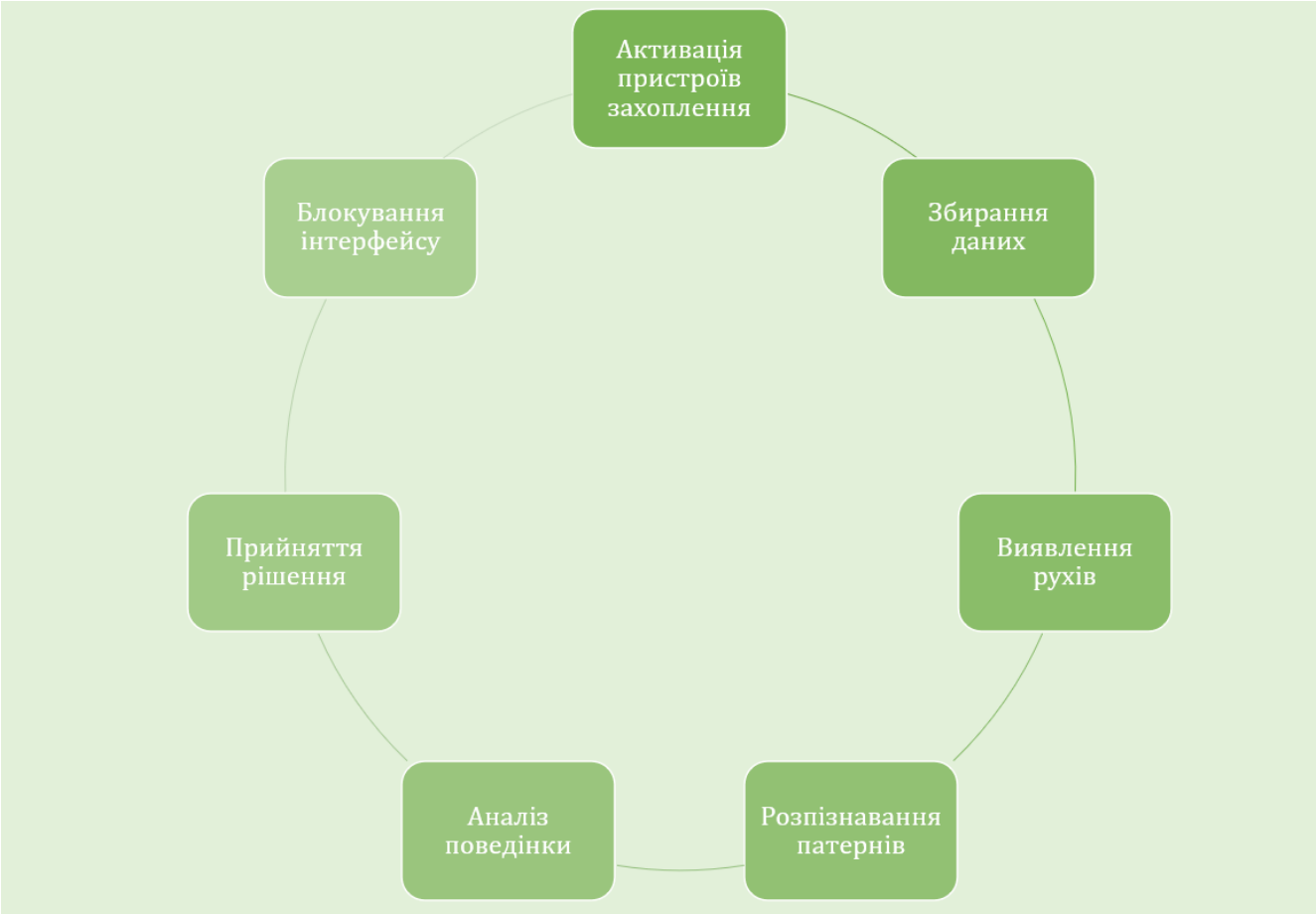


Рисунок 1.2 – Схема роботи системи motion-tracking з основними етапами

Активація сенсорів/пристроїв захоплення – на цьому етапі система активує сенсори або пристрої для захоплення рухів користувача, такі як камери, інфрачервоні датчики або акселерометри, що забезпечують точне захоплення даних про рухи.

Збирання даних – сенсори починають фіксувати дані про позицію та рухи користувача, зокрема нахили голови, положення тіла та інші жести, які є важливими для подальшого аналізу.

Виявлення рухів – система аналізує зібрані дані для ідентифікації конкретних рухів та відхилень у поведінці користувача, наприклад, часті зміни положення голови чи рук, які можуть свідчити про відволікання.

Розпізнавання патернів – на цьому етапі відбувається розпізнавання поведінкових патернів користувача. Система визначає, чи відповідають рухи типовим шаблонам поведінки для тестування або свідчать про потенційні спроби обману [15].

Аналіз поведінки – система виконує глибший аналіз виявлених патернів, оцінюючи поведінку користувача для визначення можливих ознак несанкціонованої діяльності.

Прийняття рішення – якщо система виявляє підозрілу активність, вона переходить до етапу прийняття рішення щодо реагування, яке може включати блокування інтерфейсу або сповіщення адміністратору.

Попередження/Блокування інтерфейсу – фінальний етап, на якому система виконує заплановані дії, наприклад, надсилає сповіщення або блокує доступ до інтерфейсу користувача, щоб зупинити виконання тесту у разі виявлення порушень.

Ця послідовність процесів забезпечує надійний контроль за діями користувача, підвищуючи безпеку та об’єктивність оцінювання знань і запобігаючи несанкціонованим діям під час тестування.

Реалізація відслідковування рухів і погляду вимагає використання спеціалізованого обладнання та програмного забезпечення, які дозволяють не лише зчитувати рухи, а й аналізувати дані для виявлення аномалій. Основні компоненти:

Апаратне забезпечення:

– камери з високою роздільною здатністю. Важливі для відстежування рухів очей та тіла, здатні працювати у реальному часі;

– інфрачервоні сенсори. Забезпечують високу точність зчитування рухів зіниць, навіть при низькому освітленні;

– рухомі датчики (motion sensors). Використовуються для виявлення рухів тіла користувача.

Програмне забезпечення:

– алгоритми комп'ютерного зору. Аналізують зображення та відео для визначення поведінкових патернів;

– моделі машинного навчання. Здатні розпізнавати типові аномалії у поведінці користувача та класифікувати їх за ступенем ризику;

– інтеграція з системами штучного інтелекту. Дозволяє системі постійно вчитися на поведінкових даних для вдосконалення виявлення несанкціонованих дій [15].

Системи аналізу поведінки користувача об'єднують дані відстежування рухів очей та тіла для формування комплексного профілю поведінки. Ці системи допомагають виявляти ознаки несанкціонованої активності, а саме:

– визначення профілів ризику. На основі типових патернів поведінки можна створити профіль для кожного користувача, що дозволяє швидше виявляти аномальні дії;

– детекція відхилень. Система автоматично порівнює поточну поведінку з типовою моделлю для конкретного користувача і виявляє підозрілі відхилення;

– реакція на порушення: У випадку виявлення аномалій система може автоматично блокувати інтерфейс або надсилати сповіщення адміністраторам.

Одним з основних застосувань цих технологій у системах оцінювання знань є забезпечення надійності тестування шляхом запобігання несанкціонованим діям. Для цього застосовуються такі підходи інтеграції:

– моніторинг в реальному часі. Система здійснює постійний контроль за діями користувача, що дозволяє оперативно реагувати на спроби обману;

– автоматичне блокування інтерфейсу. При виявленні підозрілої поведінки інтерфейс тестування блокується, що унеможливлює подальше виконання завдання;

– сповіщення та журналювання подій. Система фіксує всі порушення та сповіщає адміністратора для подальшого аналізу [16].

Попри численні переваги, впровадження технологій відслідковування пов'язане з певними викликами, які потребують вирішення:

– конфіденційність та етичні аспекти. Використання технологій відслідковування в освітніх системах має відповідати вимогам конфіденційності та етичним стандартам, оскільки зібрані дані є чутливими;

– технічні обмеження. Система відслідковування вимагає високоякісного обладнання та може мати труднощі з точністю в умовах низького освітлення або нестабільного Інтернет-з'єднання;

– удосконалення алгоритмів для зниження помилкових спрацьовувань. Сучасні технології мають обмеження у вигляді хибних спрацьовувань, які можуть впливати на користувацький досвід. Майбутні розробки спрямовані на покращення алгоритмів, які здатні адаптуватися до індивідуальних особливостей кожного користувача [16].

1.4 Особливості застосування моделей контролювання для оцінювання знань

Сучасні освітні системи все частіше використовують моделі контролювання для забезпечення точності та чесності процесу оцінювання знань. Моделі контролювання у цій сфері включають алгоритми та технології, які аналізують поведінку користувача, дозволяють виявляти порушення та запобігають несанкціонованій діяльності під час тестування. Розглянемо особливості, які

визначають ефективність таких моделей, зокрема їхню структуру, методи інтеграції з оцінюванням знань, переваги, обмеження та виклики.

Моделі контролювання в освітніх системах ґрунтуються на декількох ключових компонентах, які забезпечують функціональність і точність процесу моніторингу:

– системи відслідковування. Інтегровані модулі, що включають камери, сенсори руху та інші пристрої для збирання даних про дії користувача;

– алгоритми обробки даних. Комп'ютерний зір, алгоритми машинного навчання та штучний інтелект, які аналізують дані та розпізнають поведінкові патерни;

– інтерфейс користувача з функціями блокування. Інтерфейс, який забезпечує виконання тестування та може блокуватися при виявленні порушень;

– система зберігання даних і журналювання. Зберігає інформацію про дії користувача для аналізу, а також надає адміністраторам доступ до результатів і сповіщень [17].

Ця структура дозволяє забезпечити комплексний підхід до контролю дій користувача, від фіксації і аналізу до реакції на підозрілу поведінку.

Інтеграція моделей контролювання з процесом оцінювання знань відіграє ключову роль у забезпеченні об’єктивності та достовірності результатів тестування. Моделі контролювання дозволяють відстежувати дії користувача в реальному часі та реагувати на можливі спроби шахрайства, роблячи систему тестування більш прозорою і надійною. Це досягається шляхом поєднання технологій відслідковування з алгоритмами аналізу поведінки та модулем прийняття рішень. Розглянемо основні аспекти такої інтеграції.

Моніторинг є основним компонентом інтеграції, який дозволяє системі оцінювання знань контролювати дії користувача під час тестування. Система фіксує кожну дію користувача – рухи тіла, напрямок погляду, використання клавіатури чи миші, щоб визначити, чи відповідає його поведінка нормам.

– відстежування погляду (eye-tracking). Система аналізує напрямок погляду, щоб визначити, чи зосереджений користувач на тестових питаннях або ж намагається звернутися до сторонніх матеріалів;

– відстежування рухів тіла (motion-tracking). Фіксує переміщення тіла, наприклад, нахили голови або зміну пози, які можуть вказувати на спробу отримати допомогу або звернення до зовнішніх джерел;

– аналіз активності на екрані. Система також може відстежувати активність на екрані, включаючи використання браузера, перехід між вкладками або відкриття заборонених програм [18].

Моніторинг дій під час тестування дозволяє забезпечити постійний контроль, завдяки якому система може швидко виявити та припинити несанкціоновану активність.

Система використовує алгоритми штучного інтелекту для автоматичного аналізу зібраних даних та ідентифікації аномальної поведінки. Завдяки аналізу патернів поведінки система визначає, чи відповідає поведінка користувача типовим шаблонам тестування, чи містить відхилення, що можуть вказувати на порушення.

– використання машинного навчання. Алгоритми машинного навчання дозволяють системі навчатися на поведінці користувачів і виділяти типові патерни, що знижує кількість помилкових спрацьовувань і покращує точність аналізу;

– розпізнавання аномалій. Система визначає відхилення від нормальної поведінки, наприклад, часте відведення погляду за межі екрану, надмірну активність миші або клавіатури, що може свідчити про спробу шахрайства;

– розподіл рівнів ризику. Відповідно до виявлених патернів, система може встановлювати різні рівні ризику для кожної дії, що допомагає класифікувати порушення за ступенем серйозності та обирати відповідну реакцію.

Автоматичний аналіз поведінки підвищує ефективність контролю, дозволяючи швидко ідентифікувати підозрілу активність та запобігати порушенням [18].

При виявленні підозрілої активності система може автоматично ініціювати відповідні дії. Реакція на порушення залежить від рівня ризику, який система приписує певній поведінці. Вбудовані механізми реагування забезпечують ефективне припинення несанкціонованих дій.

– автоматичне блокування інтерфейсу. Якщо система виявляє високий рівень ризику, вона може автоматично заблокувати інтерфейс користувача, що призупиняє тестування. Це дозволяє уникнути продовження порушення і зберегти чесність тесту;

– надсилання сповіщень. Для порушень середнього ризику система може надсилати сповіщення адміністратору чи проктору, який відповідає за процес тестування. Це дозволяє здійснити індивідуальний контроль і визначити подальші дії;

– журналювання подій. Система автоматично фіксує всі дії користувача та відхилення від норми у журналі подій, що надає адміністраторам доступ до інформації про всі порушення для подальшого аналізу.

Механізм реагування дозволяє системі оперативно відповідати на порушення, знижуючи ризики шахрайства і забезпечуючи чесність процесу оцінювання [19].

Під час тестування всі дані, які стосуються дій та поведінки користувача, зберігаються в базі даних. Це забезпечує можливість подальшого аналізу, а також використовується для вдосконалення системи.

– збереження історії дій. Дані про всі дії та спроби порушення, зафіксовані під час тестування, зберігаються для подальшого аналізу і навчання системи;

– аналіз для вдосконалення алгоритмів. Зібрані дані використовуються для налаштування моделей контролю, вдосконалення алгоритмів розпізнавання аномалій та зменшення кількості хибних спрацьовувань;

– підтримка прозорості. Адміністратори та викладачі можуть отримати доступ до журналу дій користувача та використовувати ці дані для розгляду апеляцій або надання зворотного зв’язку.

Зберігання та аналіз даних підвищують надійність і адаптивність системи, дозволяючи налаштувати її під конкретні вимоги та особливості різних освітніх платформ.

Оскільки інтеграція моделей контролювання передбачає збирання та аналіз чутливих даних про поведінку користувачів, питання конфіденційності є критично важливим. Захист особистої інформації користувачів і запобігання несанкціонованому доступу до даних є необхідними для дотримання етичних стандартів [19].

– шифрування даних. Дані про поведінку користувача повинні зберігатися у зашифрованому вигляді для захисту від несанкціонованого доступу;

– контроль доступу. Система обмежує доступ до даних про поведінку користувачів лише для авторизованих осіб, що забезпечує збереження конфіденційності;

– використання анонімізації. Деякі дані можуть бути анонімізовані, щоб зменшити ризик розкриття особистої інформації користувачів;

– інформування та отримання згоди. Користувачі повинні бути проінформовані про використання технологій контролювання та надати свою згоду на збирання даних, що підвищує довіру до системи.

Забезпечення конфіденційності даних під час інтеграції моделей контролювання дозволяє підтримувати етичні стандарти та захищати права користувачів.

Інтеграція моделей контролювання з системою оцінювання знань дозволяє ефективно моніторити дії користувача під час тестування, виявляти та запобігати шахрайству. Це підвищує об’єктивність і достовірність результатів, забезпечуючи чесний і прозорий процес оцінювання знань у дистанційній освіті.

Використання моделей контролювання у процесі оцінювання знань надає значні переваги для освітніх закладів та учасників тестування. Ці моделі дозволяють підвищити об'єктивність, прозорість і достовірність процесу оцінювання, знижуючи ймовірність шахрайства та забезпечуючи чесне

проходження тестів. Моделі контролювання можуть аналізувати дії користувача у реальному часі, що робить процес тестування більш надійним. Основні переваги використання таких моделей включають об’єктивність, прозорість, адаптивність і попередження шахрайства, які детально розглянуті нижче [20].

Однією з головних переваг моделей контролювання є забезпечення об'єктивності під час оцінювання знань. Завдяки автоматичному моніторингу дій користувача та мінімізації людського фактору, система надає чесні й однакові умови для всіх учасників тестування. Моделі контролювання здатні автоматично фіксувати підозрілі дії, що знижує ймовірність упередженості або помилок з боку екзаменаторів.

– мінімізація людського фактору. Оскільки процес моніторингу автоматизований, викладачі або проктори не впливають на результати тестування;

– зменшення помилок. Алгоритми аналізують дії користувача на основі заздалегідь визначених критеріїв, що знижує ймовірність хибних звинувачень або пропущених порушень;

– єдині стандарти для всіх учасників. Автоматизована система забезпечує однакові умови для всіх студентів, незалежно від особистих обставин чи місця тестування, що підвищує об'єктивність оцінювання.

Моделі контролювання значно підвищують прозорість тестування, оскільки всі дії користувача фіксуються та можуть бути доступні для аналізу. Прозорість процесу дає студентам більше впевненості в неупередженості результатів і знижує ризики апеляцій та оскаржень результатів тестів [20].

– документування всіх дій користувача. Система веде детальний журнал всіх дій, які можуть бути переглянуті викладачем чи адміністрацією для подальшого аналізу;

– підтримка процесу апеляцій. У разі виникнення конфліктів або сумнівів щодо результатів тестування викладачі можуть переглянути журнал, щоб об’єктивно оцінити ситуацію;

– покращення зворотного зв’язку. Викладачі можуть надавати студентам докладний зворотний зв’язок, враховуючи збережену інформацію про їхні дії під час тесту, що робить процес навчання більш інформативним.

Інтеграція моделей контролювання у систему оцінювання знань дозволяє значно знизити ймовірність шахрайства під час тестування. Система контролю фіксує всі дії користувача в реальному часі, що ускладнює або робить неможливим виконання несанкціонованих дій [21].

– виявлення підозрілих дій. Алгоритми відстежування можуть розпізнавати такі аномалії, як відведення погляду за межі екрану, часті перемикання між вкладками чи підозрілі рухи тіла;

– автоматичне блокування інтерфейсу. Система може миттєво блокувати доступ до тесту або видавати попередження, якщо було виявлено потенційну спробу шахрайства, що допомагає припинити порушення в реальному часі;

– попередження використання зовнішніх ресурсів. Система відстежує активність на екрані та інші дії, що обмежує можливість використання сторонніх матеріалів під час тестування.

Моделі контролювання можуть адаптуватися до індивідуальної поведінки користувача, що дозволяє уникнути хибних спрацьовувань і підвищує точність аналізу. Алгоритми навчання можуть вивчати типові патерни поведінки для кожного користувача і враховувати їх у процесі оцінювання.

– адаптивні алгоритми розпізнавання. Завдяки машинному навчанню система може аналізувати поведінку різних користувачів і відрізняти індивідуальні особливості від потенційно підозрілих дій;

– зниження помилкових спрацьовувань. Адаптивні моделі знижують кількість хибних спрацьовувань, що робить тестування комфортнішим для студентів;

– аналіз поведінкових трендів. Система може виявляти патерни поведінки на основі даних з попередніх тестів, що дозволяє більш точно визначати типові реакції кожного користувача [21].

Використання моделей контролювання підвищує рівень захисту даних про користувача, оскільки система автоматично фіксує всі дії і забезпечує збереження інформації у зашифрованому вигляді. Забезпечення конфіденційності та безпеки дозволяє користувачам відчувати себе захищеними під час проходження тестів.

– шифрування даних. Всі дані зберігаються у зашифрованому вигляді, що захищає їх від несанкціонованого доступу та витоку;

– обмеження доступу до даних. Тільки авторизовані користувачі, такі як адміністратори та викладачі, мають доступ до збережених даних, що підвищує рівень безпеки;

– відповідність стандартам конфіденційності. Система контролю дотримується стандартів захисту даних, таких як GDPR, забезпечуючи захист прав студентів на конфіденційність.

Завдяки даним, зібраним під час тестування, викладачі можуть детально аналізувати поведінку студентів, виявляти слабкі місця у процесі навчання та коригувати методи викладання. Це дозволяє оптимізувати навчальний процес і підвищити ефективність освітніх програм [22].

– аналіз когнітивних навантажень. Відстеження уваги та концентрації дозволяє викладачам визначати, які завдання викликають труднощі, і коригувати навчальні програми;

– індивідуальний підхід до навчання. На основі даних про поведінку користувача можна адаптувати методику навчання під індивідуальні потреби студента;

– забезпечення зворотного зв’язку. Завдяки даним про проходження тестів, викладачі можуть надавати студентам більш обґрунтований зворотний зв’язок, що дозволяє студентам краще розуміти свої сильні та слабкі сторони.

Загалом, моделі контролювання у процесі оцінювання знань підвищують рівень об’єктивності, прозорості, чесності та безпеки під час тестування. Це дозволяє створити ефективнішу систему навчання, яка не лише запобігає порушенням, але й підтримує оптимальний навчальний процес для студентів.

Незважаючи на численні переваги моделей контролювання у процесі оцінювання знань, їх використання також супроводжується певними викликами та обмеженнями. Серед них — технічні складнощі, конфіденційність та безпека даних, можливість помилкових спрацьовувань і етичні аспекти використання. Розглянемо кожен з цих викликів і обмежень докладніше.

Одним із основних викликів для моделей контролювання є технічні обмеження, зокрема вимоги до якості апаратного та програмного забезпечення, а також стабільності Інтернет-з’єднання. Технічні обмеження можуть впливати на точність відстежування рухів користувача і знижувати ефективність системи в цілому [22].

– вимоги до обладнання. Використання моделей контролювання вимагає наявності високоякісних камер і сенсорів для точного відстежування рухів. Погана роздільна здатність або відсутність необхідного обладнання можуть обмежувати можливості системи;

– залежність від умов освітлення. Технології, що використовують камери, як-от eye-tracking і motion-tracking, можуть працювати менш точно при слабкому освітленні або відблисках, що може знижувати якість збору даних;

– залежність від стабільного Інтернет-з'єднання. Дистанційні системи контролювання часто потребують стабільного Інтернет-з'єднання для передачі даних у реальному часі. Переривання або нестабільність з'єднання можуть призвести до втрати даних або переривання тестування, що знижує ефективність контролю.

Моделі контролювання зосереджені на зборі великих обсягів особистих даних про поведінку користувача, що викликає питання щодо конфіденційності та захисту інформації. Оскільки системи моніторингу фіксують чутливі дані, необхідно забезпечити їх збереження та запобігти несанкціонованому доступу.

– захист персональних даних. Уся інформація, яка стосується поведінки користувача, має бути захищена від витоку або несанкціонованого доступу. Це

вимагає використання шифрування даних і обмеження доступу лише для авторизованих осіб;

– дотримання стандартів конфіденційності. Система має відповідати міжнародним стандартам, як-от GDPR, щоб забезпечити дотримання прав користувачів на конфіденційність і безпеку особистої інформації;

– складність зберігання великих обсягів даних. Системи контролювання повинні зберігати дані про кожну дію користувача протягом тривалого часу, що вимагає значних ресурсів для обробки і зберігання, а також забезпечення належного захисту.

Моделі контролювання можуть помилково ідентифікувати певну поведінку як порушення, що негативно впливає на досвід користувача і може призвести до непорозумінь. Помилкові спрацьовування є важливим обмеженням систем контролювання, які працюють на основі алгоритмів штучного інтелекту та машинного навчання [23].

– помилкові спрацьовування. Наприклад, невинні рухи голови або часте моргання можуть бути помилково інтерпретовані як спроба шахрайства, що може спричинити необґрунтоване блокування доступу до тесту;

– недосконаліть алгоритмів розпізнавання. Алгоритми можуть не завжди точно розпізнавати поведінку, особливо за відсутності достатньої кількості навчальних даних, що призводить до неточностей в оцінюванні;

– адаптивність до різних патернів поведінки. Індивідуальні особливості поведінки кожного користувача можуть впливати на спрацювання алгоритмів, тому система повинна бути здатною розрізняти стандартні патерни та підозрілі дії, що може бути складним завданням.

Системи контролювання під час оцінювання знань викликають етичні питання щодо ступеня втручання в особисте життя користувача. Надмірний контроль може спричинити психологічний тиск і стрес у користувачів, що негативно впливає на їхній досвід під час тестування.

– довіра до системи. Надмірне втручання в поведінку користувача може знизити рівень довіри до системи, особливо якщо студенти відчувають, що за ними надмірно стежать;

– психологічний вплив. Постійне спостереження та ризик помилкових спрацьовувань можуть спричинити стрес у користувачів, особливо під час іспитів чи інших відповідальних заходів;

– прозорість моніторингу. Користувачі повинні бути поінформовані про те, як саме працює система, що відстежується, як використовуються їхні дані та яка буде реакція у випадку виявлення порушень, щоб забезпечити дотримання етичних норм і зменшити стрес [23].

Інтеграція моделей контролювання з існуючими платформами для онлайн-освіти, такими як LMS (Learning Management System), може бути складною. Кожна освітня система має свої особливості, і для забезпечення сумісності потрібна адаптація технологій та програмного забезпечення.

– вимоги до сумісності. Різні освітні системи можуть мати різні технічні вимоги, що ускладнює інтеграцію моделей контролювання;

– високі витрати на інтеграцію. Інтеграція з існуючими системами може потребувати значних фінансових та технічних ресурсів, особливо якщо освітня платформа не передбачала інтеграцію таких технологій;

– складність оновлення та підтримки. Постійне вдосконалення алгоритмів і програмного забезпечення для забезпечення високого рівня точності та безпеки системи вимагає значних зусиль з боку розробників.

Ще одним викликом є ризик неправомірного використання даних про поведінку користувачів. Оскільки моделі контролювання збирають чутливу інформацію про кожен крок користувача під час тестування, існує ризик, що ці дані можуть бути використані для інших цілей без згоди користувача.

– ризик використання для сторонніх цілей. Зібрані дані можуть зацікавити сторонніх осіб або організації, що створює ризик порушення конфіденційності, якщо дані використовуватимуться для інших цілей;

– необхідність строгого контролю доступу. Для забезпечення конфіденційності слід обмежувати доступ до даних, але це створює додаткові виклики для адміністрування системи;

– правові аспекти. Використання поведінкових даних повинно відповідати вимогам законодавства про захист даних, і порушення цих вимог може призвести до серйозних юридичних наслідків [23].

Отже, попри значні переваги, моделі контролювання під час оцінювання знань стикаються з низкою викликів і обмежень. Для ефективного їх використання необхідно враховувати технічні, етичні, правові та конфіденційні аспекти, а також постійно вдосконалювати алгоритми, щоб мінімізувати ризики і забезпечити надійний,

1.5 Аналіз існуючих моделей контролювання дій користувача

Аналіз існуючих моделей контролювання дій користувача є важливим етапом при розробці нової системи оцінювання знань, орієнтованої на забезпечення високого рівня чесності, об'єктивності та безпеки під час тестування. Існуючі системи оцінювання знань мають різні методи контролю, що варіюються від простого фіксування активності на екрані до комплексного відеопрокторингу з безперервним моніторингом користувача. Проте, незважаючи на наявність цих моделей, вони мають певні обмеження, які впливають на їхню ефективність у виявленні шахрайства та забезпеченні чесності під час онлайн-тестування [24].

Під час аналізу існуючих моделей контролювання дій користувача розглядаються такі ключові аспекти:

– точність і ефективність виявлення шахрайства. Наскільки ефективно кожна модель визначає спроби порушення, як-от звернення до сторонніх джерел чи використання зовнішніх матеріалів;

– адаптивність до поведінки користувача. Здатність моделей розпізнавати індивідуальні патерни поведінки і відрізняти їх від підозрілих дій, що може зменшити кількість хибних спрацьовувань;

– автоматизація прийняття рішень. Можливість автоматичного блокування інтерфейсу, надсилання сповіщень або прийняття інших заходів у разі виявлення підозрілої активності;

– захист даних та конфіденційність. Наскільки система захищає персональні дані користувача і чи відповідає вона вимогам законодавства щодо конфіденційності;

– зручність для користувача. Рівень психологічного комфорту під час проходження тесту, що залежить від інвазивності моніторингу;

– витрати на впровадження та підтримку. Вартість апаратного та програмного забезпечення, необхідного для реалізації системи, а також витрати на її обслуговування.

Аналіз існуючих моделей контролювання дозволяє чітко розуміти, які аспекти кожної системи є ефективними, а які потребують вдосконалення. Існуючі системи контролю, як-от відеопрокторинг, eye-tracking або motion-tracking, мають певні переваги, але також стикаються з обмеженнями. Наприклад, відеопрокторинг забезпечує пряму перевірку в реальному часі, проте залежить від людського фактору та Інтернет-з'єднання. Аналіз цих аспектів дає змогу розробникам побачити, які сильні сторони варто зберегти, а які недоліки треба уникнути або мінімізувати в майбутній системі.

Аналіз дає змогу визначити, які інноваційні технології, такі як відстеження рухів і погляду, можуть бути корисними в рамках нової системи контролю. Це включає оцінку їхньої вартості, технічної здійсненності, потреби в додатковому обладнанні та потенційних вигод у вигляді підвищення ефективності контролю. Використання нових технологій може значно покращити точність виявлення шахрайства та зменшити потребу у постійному нагляді з боку прокторів, забезпечуючи більш автоматизований підхід до контролю [24].

Існуючі системи оцінювання знань, які мають обмежені можливості контролю, стикаються з труднощами у запобіганні шахрайству, особливо в онлайн-середовищі. Аналіз існуючих моделей контролювання дозволяє краще зрозуміти,

як можна підвищити об’єктивність оцінювання та забезпечити рівні умови для всіх користувачів. Наприклад, завдяки розширеному моніторингу за допомогою технологій відстежування рухів і погляду, можна забезпечити, щоб усі учасники тестування дотримувалися встановлених правил без винятків, що знижує ризик маніпуляцій та підвищує загальну справедливість процесу.

Проблеми з конфіденційністю і психологічний тиск є важливими аспектами, які потребують особливої уваги під час впровадження систем контролювання. Деякі існуючі моделі контролю є занадто інвазивними і можуть викликати стрес у користувачів, зокрема через постійний відеонагляд. Аналіз дозволяє оцінити, наскільки майбутня система контролю може забезпечити менший рівень втручання в особисте життя і водночас ефективний контроль. Наприклад, використання технологій, які зосереджуються на ключових показниках поведінки (погляд, рухи), може знизити інвазивність і підвищити психологічний комфорт користувачів, що сприятиме створенню довіри до системи.

Аналіз витрат та ресурсів, необхідних для підтримки існуючих систем, дозволяє краще зрозуміти економічну доцільність майбутньої розробки. Наприклад, у традиційних системах прокторингу часто потрібен людський ресурс (проктори), що збільшує вартість підтримки, тоді як автоматизовані системи можуть знижувати ці витрати в перспективі. Аналіз допомагає визначити, які інвестиції у нові технології є виправданими з точки зору довгострокової економії і підвищення ефективності контролю [25].

Цей аналіз надає основні дані для проектування і розробки нової системи, яка враховує всі аспекти — від об'єктивності й точності виявлення шахрайства до економічної доцільності і конфіденційності користувачів.

На основі проведеного аналізу можна порівняти ключові аспекти існуючих систем контролювання дій користувача та майбутньої розробки. Це порівняння дозволить чіткіше побачити, які переваги та вдосконалення може запропонувати нова система, враховуючи сучасні вимоги до забезпечення чесності та об'єктивності під час оцінювання знань. У таблиці 1.1 нижче наведено детальний

аналіз основних характеристик, що охоплюють точність контролю, адаптивність, конфіденційність, економічну доцільність та інші аспекти.

Таблиця 1.1 – Порівняльний аналіз існуючих систем контролювання дій користувача та майбутньої розробки

Критерій	Існуючі системи оцінювання знань	Майбутня розробка з вдосконаленою системою контролю
Методи контролю	Зазвичай використовують прокторинг через відеоспостереження або фіксацію активності на екрані (такі системи, як ProctorU, ExamSoft). Обмежене використання технологій відстежування рухів і погляду.	Розширене використання технологій відстежування рухів і погляду (eye-tracking, motion-tracking) для детального аналізу поведінки користувача під час тестування та автоматичного виявлення шахрайства.
Точність виявлення шахрайства	Середня. Системи фіксують основні активності на екрані або можуть виявляти відсутність уваги за допомогою відео, але обмежені у визначенні дрібних ознак несанкціонованої активності.	Висока. Завдяки технологіям eye-tracking і motion-tracking, система зможе фіксувати навіть незначні зміни в напрямку погляду, пози тіла, що дозволить точніше визначати, чи користувач відволікається на сторонні джерела інформації.
Аналіз поведінкових патернів	Обмежений аналіз. Існуючі системи зазвичай не враховують типові для кожного користувача патерни поведінки, що може призвести до помилкових спрацьовувань.	Високий рівень аналізу. Моделі, що використовуються в майбутній розробці, можуть враховувати типові поведінкові патерни кожного користувача, зменшуючи кількість помилкових спрацьовувань і забезпечуючи більш персоналізований контроль.

Продовження таблиці 1.1

Автоматизація прийняття рішень	Часткова автоматизація. Існуючі системи здебільшого покладаються на адміністратора або проктора для прийняття рішень про порушення. У разі виявлення підозрілих дій система може надіслати попередження, але не завжди автоматично реагує.	Повна автоматизація. Система здатна автоматично блокувати інтерфейс користувача або надсилати сповіщення в разі виявлення підозрілої активності, забезпечуючи швидку реакцію на потенційні порушення без необхідності втручання людини.
Захист від шахрайства з використанням зовнішніх джерел	Обмежений. Відеоспостереження не завжди забезпечує повний контроль над напрямком уваги користувача. Перемикання вкладок контролюється, але не завжди є точним маркером для виявлення шахрайства.	Підвищений рівень захисту. Завдяки eye-tracking, система може точно визначити, коли користувач відводить погляд від екрану чи намагається взаємодіяти з додатковими джерелами, що значно ускладнює спроби шахрайства.
Адаптивність до індивідуальної поведінки	Низька. Існуючі системи зазвичай не враховують індивідуальні особливості поведінки, що може викликати надмірну кількість помилкових тривог.	Висока. Модель може адаптуватися до особливостей поведінки користувача, розпізнаючи індивідуальні патерни та зменшуючи ймовірність хибних спрацьовувань.
Конфіденційність і безпека даних	Відносна. Усі основні системи прокторингу дотримуються стандартів безпеки, проте інвазивність відеоспостереження може викликати дискомфорт у користувачів.	Висока. Хоча система здійснює моніторинг за допомогою eye-tracking та motion-tracking, вона може зберігати дані у зашифрованому вигляді і забезпечувати доступ лише за необхідності, що знижує рівень стресу у користувачів.

Продовження таблиці 1.1

Витрати на впровадження та підтримку	Середні. Більшість існуючих систем прокторингу є доступними, але вони покладаються на людський ресурс (проктори), що може збільшувати вартість.	Вищі, але виправдані. Початкові витрати на впровадження технологій відстежування рухів та погляду є високими, але автоматизація процесів знижує необхідність у людському ресурсі, що в перспективі зменшує загальні витрати.
Психологічний комфорт користувача	Помірний. Деякі користувачі відчувають дискомфорт під час відеоспостереження, особливо якщо контроль здійснюється постійно.	Підвищений. Система контролює дії користувача менш інвазивно (без необхідності повного відеоспостереження), зосереджуючись на ключових індикаторах поведінки, що забезпечує більший психологічний комфорт.

Порівняння існуючих систем оцінювання знань із запропонованою вдосконаленою розробкою показує, що інтеграція технологій відстежування рухів і погляду в систему контролю дозволяє значно підвищити точність виявлення шахрайства, адаптивність до поведінки користувача та загальний рівень захисту. Майбутня розробка здатна забезпечити повну автоматизацію процесу прийняття рішень, знижуючи залежність від людського фактору і дозволяючи надійніше виявляти шахрайство.

Проте, через високу вартість впровадження і підвищені вимоги до обладнання, запропоновану систему варто використовувати в контексті високих стандартів безпеки, таких як вступні іспити або сертифікаційні екзамени, де потрібний високий рівень контролю. Водночас, для рутинного тестування чи регулярних іспитів у рамках навчання можна застосовувати існуючі рішення або комбінувати їх з елементами майбутньої розробки для забезпечення необхідного рівня безпеки при мінімальних витратах [25].

1.6 Висновки та постановка задачі

В рамках даного розділу було проведено дослідження сучасних підходів до оцінювання знань та моделей контролювання дій користувача у системах тестування. Для цього було ретельно проаналізовано різні аспекти забезпечення чесності та об’єктивності під час тестування, які включають важливість використання сучасних технологій відстежування рухів та погляду, автоматизацію прийняття рішень, адаптивність моделей контролю до індивідуальної поведінки користувача, вимоги до конфіденційності даних та збереження психологічного комфорту користувачів. Це дослідження дало змогу глибше зрозуміти і оцінити критичні аспекти, які є важливими для розробки ефективної системи оцінювання знань у дистанційному навчанні та сертифікаційних екзаменах.

Отже, на основі проведеного дослідження та аналізу було сформульовано ряд завдань для подальшої розробки:

– розробка алгоритму роботи вдосконаленої системи контролю дій користувача з інтеграцією технологій відстежування рухів та погляду;

– проектування моделі автоматизованого прийняття рішень на основі аналізу поведінки користувача, що дозволить оперативно реагувати на підозрілі дії;

– інтеграція адаптивних алгоритмів машинного навчання для зниження кількості хибних спрацьовувань та підвищення точності виявлення несанкціонованих дій;

– розробка схем захисту та шифрування даних для забезпечення конфіденційності та безпеки інформації про користувача;

– практична реалізація та тестування розробленої системи з метою підтвердження її ефективності у реальних умовах та подальшої адаптації до різних освітніх платформ.

Ці завдання спрямовані на створення комплексної та надійної системи оцінювання знань, яка зможе забезпечити високий рівень об’єктивності, прозорості та безпеки під час тестування.

2 ПРОЕКТУВАННЯ ВДОСКОНАЛЕНОЇ СИСТЕМИ ОЦІНЮВАННЯ ЗНАНЬ НА ОСНОВІ ВДОСКОНАЛЕНОЇ МОДЕЛІ КОНТРОЛЮВАННЯ ДІЙ КОРИСТУВАЧА З ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ ВІДСЛІДКОВУВАННЯ РУХІВ ТА ПОГЛЯДУ КОРИСТУВАЧА

В даному розділі буде спроектовано вдосконалену систему оцінювання знань, що базується на інтеграції технологій відстежування рухів і погляду користувача із моделями контролювання дій. Розглянуто особливості вдосконалення існуючих систем оцінювання знань через впровадження інноваційних підходів до моніторингу поведінки користувача.

Описано особливості застосування технологій відслідковування рухів і погляду в контексті моніторингу дій користувача, розроблено алгоритм роботи вдосконаленої системи контролю, а також спроєктовано модель автоматизованого прийняття рішень на основі аналізу поведінкових характеристик. У висновках сформульовано результати досліджень та визначено основні підходи до реалізації системи.

2.1 Особливості вдосконалення системи оцінювання знань на основі вдосконаленої моделі контролювання дій користувача

Підвищення точності виявлення шахрайства є одним з головних пріоритетів вдосконаленої системи контролю дій користувача в процесі тестування. Інтеграція технологій відстежування рухів (motion-tracking) і погляду (eye-tracking) дозволяє значно розширити можливості системи з точки зору точності й швидкості виявлення несанкціонованих дій. Завдяки таким технологіям система може фіксувати навіть незначні відхилення від звичайної поведінки користувача, що забезпечує надійний захист від шахрайства та підвищує об’єктивність результатів тестування [26].

Основні механізми підвищення точності виявлення шахрайства за допомогою вдосконаленої системи контролю включають такі аспекти:

1. Визначення напрямку уваги користувача.

Технологія відстежування погляду (eye-tracking) дозволяє системі точно визначити, куди спрямований погляд користувача в будь-який момент під час тестування. Це надає можливість фіксувати навіть короткочасні відхилення уваги від тестового екрана, які можуть свідчити про звернення до сторонніх джерел або спробу отримати допомогу. Система здатна:

– аналізувати напрямок і фокус погляду. Фіксуючи напрямок погляду, система може відстежувати, чи користувач дійсно зосереджений на тестових питаннях. Наприклад, часті відведення погляду в сторону або переміщення очей на інші предмети може сигналізувати про можливість шахрайства;

– фіксувати тривалість зосередження на окремих елементах екрану. Довготривале зосередження погляду на одному питанні може свідчити про труднощі або пошук відповіді у зовнішніх джерелах, тоді як часте перемикання між елементами може бути ознакою розпорошеної уваги і пошуку інформації ззовні;

– реагувати на нетипові рухи очей: Раптове відведення погляду від екрану або нестандартна траєкторія погляду можуть бути зафіксовані як підозріла поведінка, що активує додатковий контроль.

2. Виявлення фізичних рухів, що можуть свідчити про шахрайство.

Використання технології motion-tracking дозволяє контролювати рухи користувача під час тестування. Це включає фіксацію рухів голови, рук, зміни пози тіла, що може свідчити про відволікання або спробу переглядати сторонні матеріали. Така функціональність дозволяє:

– реєструвати нахили голови або зміну позиції. Система відстежує, чи користувач намагається подивитися поза межі екрану, наприклад, на телефон чи шпаргалку. Нахили голови або зміна пози можуть сигналізувати про звернення до додаткових джерел;

– фіксувати рухи рук або взаємодію з іншими предметами: Якщо користувач взаємодіє з клавіатурою, мишею або екраном у неочікуваний момент, система може зареєструвати це як можливе порушення;

– аналізувати типові і нетипові патерни рухів. Завдяки накопиченим даним система розпізнає типові патерни поведінки користувачів під час тестування і виявляє ті рухи, що не відповідають цим шаблонам [26].

1. Автоматичне блокування інтерфейсу при підозрілих діях.

Інтеграція технологій відстежування рухів та погляду дозволяє системі автоматично реагувати на підозрілі дії користувача. Система може автоматично припинити тестування у разі виявлення потенційного порушення, що значно ускладнює спроби шахрайства та забезпечує оперативний контроль:

– миттєве блокування: Якщо система фіксує серію підозрілих дій, таких як відведення погляду або нахил голови, вона може призупинити тестування і вимагати підтвердження готовності продовжити;

– надсилання сповіщень адміністратору. У разі виявлення підозрілої активності система може автоматично надіслати сповіщення проктору або адміністратору для негайного втручання;

– журналювання інцидентів. Усі зафіксовані підозрілі дії записуються у журналі подій, що дозволяє зберегти історію порушень і надає адміністраторам можливість подальшого аналізу.

2. Зменшення кількості хибних спрацьовувань.

Використання алгоритмів машинного навчання дозволяє системі аналізувати і адаптуватися до типових патернів поведінки користувачів, що значно знижує кількість хибних спрацьовувань. Це забезпечує високу точність і підвищує загальну надійність системи:

– аналіз типових патернів поведінки: Алгоритми розпізнають звичні рухи користувачів, що дозволяє уникати випадкових спрацьовувань на нешкідливі дії, наприклад, часте моргання або випадкові рухи тіла;

– навчання на індивідуальній поведінці: Система здатна враховувати унікальні патерни поведінки кожного користувача, що дозволяє підвищити точність і зменшити рівень стресу користувачів під час тестування;

– точне налаштування чутливості. Алгоритми машинного навчання дозволяють налаштовувати рівень чутливості до рухів і погляду для кожного

окремого тестування, що підвищує об’єктивність контролю і знижує психологічне навантаження на користувачів [27].

Автоматизація процесу прийняття рішень є важливою складовою вдосконаленої системи контролювання дій користувача, що забезпечує швидке і об’єктивне реагування на підозрілі дії без необхідності втручання людини. Впровадження автоматизації дозволяє знизити залежність від людського фактору, скоротити час реакції на потенційні порушення та підвищити загальну об'єктивність і точність оцінювання. Завдяки алгоритмам, які інтегруються з технологіями відстежування рухів та погляду, система може самостійно ухвалювати рішення про підозрілі дії та відповідно реагувати в реальному часі.

Основні компоненти автоматизації процесу прийняття рішень у вдосконаленій системі контролювання дій користувача включають:

Миттєве розпізнавання та класифікація підозрілих дій.

Система використовує алгоритми машинного навчання та обробки даних у реальному часі для швидкого розпізнавання та класифікації підозрілих дій користувача. Це дозволяє системі негайно ідентифікувати потенційно несанкціоновану активність і оцінити її серйозність.

– аналіз і класифікація патернів. Алгоритми здатні розпізнавати стандартні та аномальні патерни поведінки, такі як часте відведення погляду, нахили голови або взаємодія з іншими об’єктами під час тестування;

– класифікація за рівнем ризику. Кожна підозріла дія отримує рівень ризику (низький, середній або високий), що визначає подальші дії системи – від простого попередження до негайного блокування інтерфейсу;

– самонавчання та покращення. Завдяки машинному навчанню система з часом удосконалюється, запам’ятовуючи індивідуальні особливості поведінки користувачів, що зменшує кількість хибних спрацьовувань і підвищує точність класифікації [27].

Автоматичне блокування інтерфейсу користувача.

Одним з головних завдань автоматизованої системи є можливість миттєвого блокування інтерфейсу користувача у випадку виявлення серії високоризикових

дій. Ця функція забезпечує негайне припинення тестування у разі серйозного порушення, унеможливлюючи подальше списування або отримання сторонньої допомоги.

– миттєва реакція на порушення. При виявленні серії підозрілих дій високого рівня ризику (наприклад, різке відведення погляду та тривалі нахили голови), система автоматично блокує інтерфейс користувача, припиняючи доступ до тесту;

– тимчасове блокування з можливістю поновлення. Система може налаштовуватися на тимчасове блокування з можливістю поновлення після короткої перевірки, що дозволяє уникати непотрібного стресу у випадку незначних порушень;

– журналювання інцидентів. Всі дії користувача під час автоматичного блокування записуються в журнал подій, що забезпечує докладну документацію для подальшого аналізу або розгляду апеляцій [28].

Надсилання автоматичних сповіщень адміністраторам.

При виявленні підозрілих дій система може автоматично надсилати сповіщення адміністратору або проктору, що дозволяє йому здійснити індивідуальний контроль та оцінити ситуацію. Це знижує навантаження на адміністратора, дозволяючи йому реагувати тільки в критичних ситуаціях.

– гнучкі параметри сповіщень. Система може бути налаштована на надсилання сповіщень лише за умов високого ризику або при повторюваних підозрілих діях, що забезпечує оптимальний рівень автоматизації;

– докладний опис інциденту. Кожне сповіщення містить інформацію про зафіксовані дії та рівень ризику, що дає адміністратору змогу швидко зрозуміти характер порушення та вжити відповідних заходів;

– налаштування сповіщень у реальному часі. Адміністратори можуть отримувати сповіщення безпосередньо під час тестування, що дозволяє забезпечити швидке реагування на потенційні порушення;

– інтелектуальне управління ризиками.

Система може використовувати складні алгоритми для управління ризиками та оцінки дій користувача в реальному часі, забезпечуючи об’єктивність і точність

під час прийняття рішень. Інтелектуальні алгоритми дозволяють системі визначати ступінь ризику кожної дії та відповідно коригувати рівень контролю.

– аналіз ризиків у режимі реального часу. На основі даних про поведінку користувача, система динамічно оцінює ризики та приймає рішення про необхідність втручання, що знижує ймовірність помилкових дій;

– корекція рівня контролю залежно від ризику. Залежно від поведінки користувача, система може автоматично знижувати або підвищувати рівень контролю, що забезпечує гнучкість і адаптивність до індивідуальних особливостей;

– побудова профілю користувача. Система створює поведінковий профіль кожного користувача на основі попередніх тестів, що підвищує точність визначення підозрілих дій та знижує рівень стресу для користувача [28].

Зниження залежності від людського фактору та забезпечення об’єктивності.

Завдяки автоматизації процесу прийняття рішень система забезпечує об’єктивний підхід до контролю за користувачами, мінімізуючи суб’єктивні помилки. Це дозволяє зробити оцінювання знань більш прозорим і надійним, знижуючи вплив людського фактору.

– уникнення суб’єктивних оцінок. Автоматизована система контролює дії користувача на основі заздалегідь визначених алгоритмів, що знижує можливість суб’єктивних помилок та упередженості;

– однакові умови для всіх користувачів. Автоматизація забезпечує рівні умови контролю для всіх користувачів, незалежно від часу чи обставин тестування;

– зниження навантаження на прокторів. Завдяки автоматизації проктори можуть втручатися тільки в критичних випадках, що знижує їхнє навантаження та дозволяє ефективніше використовувати ресурси.

Адаптивність до індивідуальних поведінкових патернів є важливою характеристикою вдосконаленої системи контролювання дій користувача, оскільки вона дозволяє системі пристосовуватися до унікальної поведінки кожного користувача, що підвищує точність виявлення підозрілих дій і знижує кількість хибних спрацьовувань. Індивідуальні патерни поведінки користувачів можуть включати звичні рухи, характер моргання, частоту відведення погляду та інші

фізичні особливості, які не обов'язково є ознаками шахрайства, але можуть викликати спрацьовування в системах без адаптивності [29].

Завдяки алгоритмам машинного навчання та індивідуальному налаштуванню, вдосконалена система контролю може автоматично розпізнавати особливості кожного користувача та адаптувати свої реакції, підвищуючи об'єктивність оцінювання.

Аналіз і навчання на основі поведінкових патернів.

Система використовує дані про поведінку кожного користувача для створення його індивідуального профілю, який допомагає відрізнити звичайну поведінку від підозрілих дій. Це забезпечує точніше розпізнавання порушень і підвищує надійність системи.

– створення індивідуальних профілів користувачів. Система збирає дані про поведінку користувача, включаючи тривалість погляду на екран, частоту моргання, нахили голови та інші параметри, формуючи унікальний профіль;

– врахування індивідуальних патернів. На основі накопичених даних система виділяє типові для кожного користувача патерни поведінки, що дозволяє уникнути помилкових спрацьовувань на індивідуальні особливості, такі як часті моргання через втому чи звичка часто змінювати позу;

– адаптація до змін у поведінці. У процесі проходження тесту або з часом, система здатна адаптуватися до змін у поведінці користувача, що дозволяє зменшити кількість спрацьовувань через нетипові для користувача дії, спричинені стресом або фізичними умовами.

Зменшення хибних спрацьовувань.

Завдяки здатності системи навчатися на основі індивідуальних патернів поведінки, знижується кількість хибних спрацьовувань, що забезпечує користувачеві більш комфортне середовище для проходження тестування. Це особливо важливо в умовах високого навантаження та стресу, коли звичайні дії користувача можуть відрізнятися від стандартних [29].

- аналіз типових і нетипових дій. Система може визначати, які дії є типовими для користувача, і ігнорувати їх під час моніторингу. Наприклад, якщо користувач

часто відводить погляд в сторону на короткий час, це не буде сприйматися як спроба шахрайства;

- точне налаштування параметрів чутливості. Виходячи з індивідуальних патернів, система може автоматично налаштовувати рівень чутливості для кожного користувача, що дозволяє зменшити ймовірність помилкового реагування на звичайні рухи;

- зменшення рівня стресу. Мінімізація кількості попереджень і хибних блокувань допомагає знизити рівень стресу у користувачів, що сприяє створенню сприятливих умов для об'єктивного оцінювання знань.

Покращення точності виявлення підозрілої активності.

Адаптивність дозволяє системі фокусуватися на справжніх порушеннях, відрізняючи їх від поведінкових особливостей користувача, які не свідчать про шахрайство. Це значно покращує точність контролю та ефективність моніторингу.

- аналіз нетипових патернів. Система здатна визначати, коли поведінка користувача відрізняється від його звичних патернів, і тільки тоді активувати процес прийняття рішення про підозрілу активність;

- автоматичне оновлення поведінкових моделей. На основі поведінкових даних кожного користувача система може оновлювати моделі розпізнавання шахрайства, враховуючи нові дані та покращуючи свої алгоритми для точнішого виявлення порушень;

- реакція на реальні аномалії. Завдяки адаптації до індивідуальної поведінки, система швидше реагує на справжні аномалії, ігноруючи звичайні для конкретного користувача дії, що підвищує надійність системи [30].

Застосування машинного навчання для адаптації.

Завдяки машинному навчанню вдосконалена система може безперервно навчатися на основі індивідуальних дій користувача та автоматично підлаштовуватися до його поведінкових особливостей, що значно покращує загальну точність і ефективність системи контролю.

- безперервне навчання і вдосконалення. Алгоритми машинного навчання дозволяють системі безперервно оновлювати знання про кожного користувача, що підвищує точність контролю і знижує кількість помилок;

- розпізнавання змін у поведінці. Система здатна розпізнавати зміни у поведінці користувача, такі як зниження концентрації через втому, що дозволяє уникати хибних спрацьовувань;

- індивідуальний підхід до контролю. Використання алгоритмів машинного навчання допомагає створити індивідуалізовані моделі контролю для кожного користувача, підвищуючи точність виявлення порушень [30].

2.2 Особливості використання технології відслідковування рухів та погляду користувача

Технологія відслідковування погляду (eye-tracking) є сучасним інструментом, який дозволяє фіксувати рухи та напрямок погляду користувача в реальному часі. Eye-tracking забезпечує можливість аналізувати, куди саме спрямований погляд користувача, як довго він зосереджується на певних елементах екрану, а також як змінюється його фокус під час тестування. У системах контролю знань ця технологія є особливо корисною для виявлення підозрілих дій, пов’язаних із відволіканням або спробами звернення до сторонніх джерел інформації.

Основні компоненти технології eye-tracking включають спеціальні камери та інфрачервоні сенсори, які фіксують рухи очей та визначають точку фіксації погляду на основі зображення райдужної оболонки. Це дозволяє з високою точністю виявляти напрямок і переміщення погляду.

Основні функції та можливості eye-tracking у контролюванні дій користувача

Eye-tracking забезпечує кілька ключових функцій, які підвищують ефективність системи контролю під час тестування, а саме:

- визначення фокусу уваги. Eye-tracking дозволяє точно визначати, на якій частині екрану зосереджений погляд користувача. Це дозволяє оцінити, чи

користувач переглядає саме тестові питання, чи ж його увага спрямована на інші об'єкти, що можуть бути джерелами допомоги або шпаргалок;

- фіксація тривалості зосередження на окремих елементах. Технологія відстежування погляду дозволяє виміряти тривалість фокусування на конкретних елементах екрану. Це дає змогу виявляти, які питання викликають у користувача найбільше труднощів або де він затримується непропорційно довго, що може свідчити про спроби знайти відповіді в зовнішніх джерелах;

- виявлення підозрілих відведень погляду. Технологія фіксує будь-які відхилення погляду за межі екрану або на певні зони, що можуть свідчити про відволікання або пошук сторонньої допомоги. Часте відведення погляду в бік може бути ознакою того, що користувач звертається до додаткових матеріалів, наприклад, телефону або нотаток.

Принципи роботи eye-tracking

Eye-tracking використовує комбінацію спеціальних камер та інфрачервоних сенсорів, які фіксують рухи очей користувача. Основні принципи роботи включають:

- виявлення райдужної оболонки та зіниці. Камери фіксують розташування райдужної оболонки та положення зіниці, що дозволяє точно визначити напрямок погляду;

- трекінг положення очей відносно екрану. Положення очей користувача фіксується у відношенні до екрану, що дозволяє виявляти зони екрану, на які спрямована увага, або ж відведення погляду за його межі;

- обробка даних у реальному часі. Зібрані дані обробляються миттєво, що дозволяє системі реагувати на будь-які відхилення в поведінці, включаючи часте або тривале відведення погляду від екрану.

Переваги eye-tracking для контролю під час тестування.

Інтеграція eye-tracking у системи контролю за тестуванням забезпечує ряд переваг, що підвищують об'єктивність і надійність оцінювання:

- миттєве виявлення підозрілих дій. Система може швидко виявляти потенційні порушення, такі як відведення погляду від екрану, і автоматично

реагувати на них, наприклад, надсилаючи попередження або блокуючи доступ до тесту;

- адаптивність до різних типів поведінки. Завдяки можливостям збирання індивідуальних даних про погляд кожного користувача, система може налаштовуватися на певні особливості користувачів, наприклад, звички відводити погляд, що знижує кількість хибних спрацьовувань;

- підвищення об'єктивності оцінювання. Eye-tracking дозволяє уникати суб'єктивності в оцінці поведінки користувача, оскільки система автоматично фіксує всі переміщення погляду, а отже, забезпечує однакові умови контролю для всіх учасників тестування.

Виклики та обмеження технології eye-tracking.

Незважаючи на численні переваги, eye-tracking має певні обмеження та виклики, що впливають на її ефективність і точність у контролюванні дій користувача:

- вимоги до якості обладнання. Eye-tracking вимагає високоякісних камер та інфрачервоних сенсорів для точного фіксування положення очей. Використання обладнання низької якості може знижувати точність моніторингу;

- залежність від умов освітлення. Зміни у зовнішньому освітленні, відблиски або тіні можуть негативно впливати на точність трекінгу погляду, що може призводити до помилкових спрацьовувань або зниження загальної точності;

- вплив на психологічний комфорт користувача. Постійне відстежування погляду може створювати психологічний дискомфорт у користувачів, оскільки користувачі можуть відчувати стрес через безперервний моніторинг.

Технологія відслідковування рухів (motion-tracking) є ефективним інструментом для фіксації і аналізу фізичних рухів користувача під час тестування, що дозволяє виявляти підозрілу поведінку і забезпечувати об'єктивність процесу оцінювання знань. Motion-tracking може фіксувати зміни положення голови, рук і тіла користувача, а також реєструвати аномальні рухи, які можуть свідчити про спроби відволікання або звернення до зовнішніх джерел інформації. Ця технологія

значно підвищує точність контролю та забезпечує своєчасну реакцію на можливі порушення.

Motion-tracking працює завдяки комбінації камер високої роздільної здатності та спеціальних сенсорів, які відстежують положення та рухи користувача у режимі реального часу. Це дозволяє системі моніторити поведінку користувача безперервно, автоматично реагуючи на відхилення від нормальної поведінки.

Основні функції та можливості motion-tracking у контролюванні дій користувача.

Технологія motion-tracking виконує кілька ключових функцій для забезпечення об’єктивного контролю під час тестування:

- фіксація положення голови. Система відстежує нахили та повороти голови користувача, що дозволяє визначити, чи залишається користувач зосередженим на екрані, або ж відводить погляд убік для звернення до зовнішніх матеріалів, таких як шпаргалки або додаткові пристрої;

- аналіз рухів рук і взаємодії з іншими об’єктами. Motion-tracking дозволяє фіксувати рухи рук, особливо ті, що спрямовані на зовнішні об’єкти. Наприклад, система може виявити, коли користувач піднімає руку до телефону або інших пристроїв, що може свідчити про спробу звернення до сторонніх джерел;

- моніторинг загального положення тіла. Система може визначати зміни у позиції тіла користувача, такі як різкі нахили, повороти або інші рухи, що можуть сигналізувати про спробу відволіктися або звернутися до додаткових матеріалів. Раптові або часті зміни положення тіла можуть сигналізувати про несанкціоновану поведінку.

Принципи роботи motion-tracking.

Motion-tracking працює за рахунок аналізу даних, отриманих з камер та сенсорів, які визначають положення користувача у просторі. Основні етапи роботи технології включають:

- захоплення зображень і рухів. Камери фіксують положення голови, рук та інших частин тіла користувача. Ці дані обробляються для створення тривимірної моделі користувача в режимі реального часу;

- аналіз патернів рухів. Система використовує алгоритми для розпізнавання стандартних і нетипових рухів користувача. Звичайні дії, такі як змінення пози, не будуть викликати реакцію системи, однак раптові рухи або пози, що вказують на звернення до зовнішніх матеріалів, викличуть відповідну реакцію;

- фіксація та реагування на аномальні рухи. Якщо система визначає певні рухи як аномальні або підозрілі, вона може автоматично реагувати, наприклад, надіславши попередження проктору або заблокувавши інтерфейс користувача.

Переваги motion-tracking для контролю під час тестування.

Використання motion-tracking у процесі оцінювання знань забезпечує численні переваги:

- підвищена точність виявлення порушень. Motion-tracking дозволяє фіксувати навіть найменші рухи користувача, що може свідчити про порушення. Це підвищує точність моніторингу та зменшує ймовірність пропущених випадків несанкціонованої активності;

- мінімізація суб'єктивності. Оскільки система автоматично визначає підозрілі рухи на основі заздалегідь заданих параметрів, ризик суб'єктивних помилок знижується, що забезпечує рівні умови для всіх учасників тестування;

- швидка реакція на підозрілу активність. Система може автоматично блокувати тест або надсилати сповіщення адміністратору, що дозволяє швидко реагувати на спроби порушення без затримок.

Обмеження та виклики motion-tracking.

Як і будь-яка інша технологія, motion-tracking має певні обмеження, які необхідно враховувати при її інтеграції в системи оцінювання знань:

- вимоги до якості та стабільності обладнання. Технологія motion-tracking потребує високоякісного обладнання, здатного забезпечувати точне відстежування рухів користувача. Низька роздільна здатність камери або нестабільне освітлення можуть значно знижувати точність контролю;

- чутливість до навколишніх умов. Надмірні рухи або низький рівень освітлення можуть заважати нормальній роботі motion-tracking. Деякі дії,

наприклад, виправлення положення рук або змінення пози, можуть помилково інтерпретуватися як порушення;

- вплив на психологічний комфорт користувача. Постійне відстежування рухів може викликати дискомфорт у користувачів, особливо якщо система спрацьовує на звичайні рухи. Це може створювати додатковий стрес під час тестування.

2.3 Розробка алгоритму роботи вдосконаленої системи контролю дій користувача з інтеграцією технологій відстежування рухів та погляду

Для забезпечення об'єктивного контролю під час тестування, вдосконалена система використовує алгоритм, що об'єднує технології відстежування рухів (motion-tracking) і погляду (eye-tracking). Це дозволяє виявляти підозрілу поведінку в режимі реального часу, забезпечуючи чесність процесу оцінювання знань. Алгоритм охоплює кілька основних етапів: збір даних, обробка і аналіз, класифікація, ухвалення рішення та реакція на потенційне порушення. Нижче описані основні етапи та функції алгоритму, що забезпечують ефективність і точність контролю.

Основні етапи майбутнього алгоритму:

Збір даних про погляд та рухи користувача.

- відстежування погляду. Камери та сенсори фіксують напрямок, тривалість фокусування та переміщення погляду користувача. Це дозволяє виявляти відхилення від нормальної поведінки, такі як часте відведення погляду від екрану або зосередження на сторонніх об’єктах.

- відстежування рухів тіла. Сенсори motion-tracking збирають дані про позицію голови, рук і тіла користувача, фіксуючи всі рухи, що можуть вказувати на можливе звернення до сторонніх джерел (наприклад, нахил голови в бік телефону).

Обробка та аналіз зібраних даних.

- попередня обробка даних. Отримані дані проходять фільтрацію, яка дозволяє усунути зайві шуми (наприклад, випадкові рухи, не пов’язані з тестуванням). Це підвищує точність системи;

- аналіз патернів. Алгоритм аналізує патерни поведінки користувача, порівнюючи їх з попередньо встановленими зразками (нормативними патернами), що дозволяє виявити аномальні дії.

Класифікація підозрілих дій за рівнем ризику:

- ідентифікація підозрілих дій. Алгоритм виявляє підозрілі дії, такі як відведення погляду або раптові рухи тіла, і класифікує їх за рівнем ризику (низький, середній, високий) на основі інтенсивності, тривалості та частоти таких дій;

- присвоєння рівня ризику. Рівень ризику кожної дії визначає подальші дії системи. Наприклад, підозрілі дії середнього рівня ризику можуть активувати попередження, а високого рівня — призвести до блокування тесту.

Прийняття рішення та реакція системи:

- автоматичне реагування. У разі виявлення дій високого ризику система може автоматично заблокувати доступ до тесту, призупинити його або активувати додатковий моніторинг за користувачем;

- попередження користувача. Система може надсилати користувачеві попередження у разі виявлення серії підозрілих дій, що дає змогу припинити порушення без переривання тесту;

- сповіщення адміністратора. При серйозних порушеннях система автоматично надсилає сповіщення адміністратору або проктору, які можуть втрутитися в процес у разі потреби.

Журналювання та подальший аналіз:

- запис подій у журналі. Всі дії користувача, виявлені як потенційно підозрілі, записуються в журнал подій для подальшого аналізу. Це забезпечує доказову базу у разі апеляцій або розгляду суперечок;

- аналіз поведінки для навчання алгоритму. Дані про поведінку користувачів використовуються для навчання алгоритмів, що дозволяє підвищити точність системи у майбутньому та знизити ймовірність хибних спрацьовувань.

Основні функції алгоритму:

- реагування в реальному часі. Завдяки збору та обробці даних у режимі реального часу алгоритм здатний миттєво реагувати на підозрілу активність, забезпечуючи своєчасний контроль;

- адаптація до індивідуальної поведінки користувача. Алгоритм враховує індивідуальні особливості поведінки кожного користувача, що знижує ймовірність хибних спрацьовувань. Це досягається шляхом аналізу типових патернів рухів і погляду кожного користувача;

- прозорість і доступність для адміністратора. Адміністратори можуть переглядати записи дій користувача та результати роботи алгоритму, що сприяє прозорості і забезпечує об’єктивність процесу тестування.

На основі даної інформації розробимо алгоритм (рис.2.1).

Розберемо покрокова даний алгоритм:

1. Ініціалізація системи та перевірка обладнання.

Система запускається, перевіряє наявність та справність необхідних сенсорів, камер, інтернет-з’єднання. Це забезпечує коректну роботу пристроїв та знижує ймовірність помилок під час тестування.

2. Ідентифікація користувача.

Система проводить аутентифікацію користувача через обліковий запис, а також верифікацію обличчя для запобігання підміни особистості.

3. Калібрування сенсорів та камер.

Система просить користувача виконати декілька простих завдань, таких як фокусування на певних точках екрану, для калібрування сенсорів і камер. Це забезпечує високу точність при подальшому відстежуванні рухів та погляду.

4. Встановлення початкових поведінкових параметрів.

На основі декількох хвилин моніторингу система збирає початкові дані про типові патерни поведінки користувача, які використовуються для подальшого порівняння.

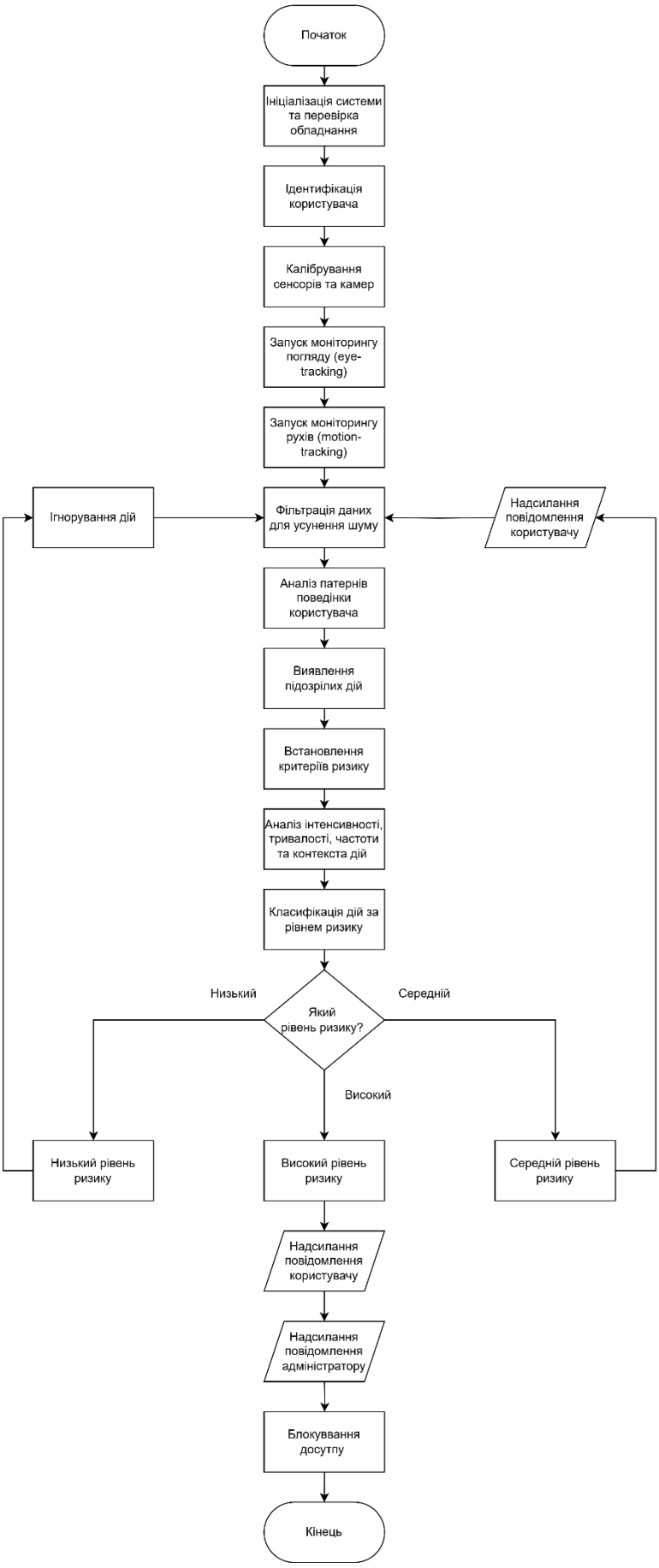


Рисунок 2.1 – Алгоритм роботи вдосконаленої системи контролю дій користувача

5. Запуск моніторингу погляду (eye-tracking).

Камери фіксують положення очей і напрямок погляду користувача, реєструючи будь-яке відведення погляду від екрану або зосередження на нецільових зонах.

6. Запуск моніторингу рухів (motion-tracking).

Сенсори фіксують положення голови, рук та загальне положення тіла. Це дозволяє визначити будь-які спроби користувача змінити позицію для звернення до сторонніх джерел.

7. Збір даних у режимі реального часу.

Система безперервно отримує і зберігає дані про напрямок погляду, рухи голови та позицію тіла користувача.

8. Фільтрація даних для усунення шуму.

Алгоритм відсіює незначні рухи та випадкові відхилення (наприклад, незначні моргання або дрібні зміни позиції), щоб уникнути зайвих спрацьовувань.

9. Аналіз патернів поведінки користувача.

Система аналізує зібрані дані, порівнюючи їх з початковими параметрами та типовими патернами поведінки для даного користувача.

10. Виявлення підозрілих дій.

Алгоритм розпізнає дії, що не відповідають типовим патернам, наприклад, часте або довготривале відведення погляду, значні нахили голови або раптові рухи рук.

11. Класифікація дій за рівнем ризику.

Система класифікує підозрілі дії за рівнем ризику (низький, середній, високий) на основі інтенсивності та частоти. Це дозволяє відрізняти незначні відхилення від серйозних порушень.

12. Реакція на дії середнього ризику: попередження користувача.

Якщо система фіксує дію середнього ризику (наприклад, коротке відведення погляду), вона надсилає користувачеві попередження для запобігання подальшому порушенню.

13. Реакція на дії високого ризику: призупинення тестування.

У разі виявлення дій високого ризику (наприклад, часте відведення погляду або взаємодія з зовнішніми об'єктами) система призупиняє тестування та блокує інтерфейс користувача.

14. Надсилання сповіщення адміністратору.

При виявленні дій високого ризику система автоматично надсилає сповіщення адміністратору або проктору, який може втрутитися для подальшої перевірки.

15. Запис дій у журналі подій.

Усі підозрілі дії, попередження та реакції системи записуються в журналі подій, що забезпечує доказову базу для подальшого аналізу.

16. Аналіз підозрілих дій та адаптація до поведінки користувача.

Якщо система виявляє багато хибних спрацьовувань, вона адаптується, оновлюючи поведінкові патерни для конкретного користувача, що знижує ймовірність хибних тривог у майбутньому.

17. Постійне навчання алгоритму.

Зібрані дані використовуються для навчання алгоритмів розпізнавання патернів та покращення точності системи в подальших сесіях.

18. Завершення тестування та оцінка результатів.

Після завершення тесту система проводить остаточний аналіз, перевіряє, чи були зафіксовані будь-які порушення, і надає адміністраторам доступ до журналу подій.

2.4 Проектування моделі автоматизованого прийняття рішень на основі аналізу поведінки користувача

Проектування моделі автоматизованого прийняття рішень є ключовим аспектом вдосконаленої системи контролю дій користувача під час тестування. Така модель дозволяє системі автоматично реагувати на підозрілу поведінку без участі людини, знижуючи залежність від людського фактору та забезпечуючи високу об'єктивність процесу. Ця модель включає збір даних про поведінку

користувача, їхній аналіз у режимі реального часу, класифікацію за рівнями ризику, ухвалення рішень на основі визначених правил і адаптивне навчання алгоритму.

Основні компоненти моделі автоматизованого прийняття рішень охоплюють такі етапи:

– Збір даних та попередня обробка.

Перед тим як ухвалити будь-яке рішення, система збирає дані про поведінку користувача за допомогою технологій відстежування рухів (motion-tracking) та погляду (eye-tracking). На цьому етапі фіксуються такі параметри, як напрямок і тривалість фокусування погляду, рухи голови, рук, зміни позиції тіла. Попередня обробка даних дозволяє відфільтрувати несуттєві шуми та випадкові дії.

– Аналіз поведінкових патернів користувача.

Модель використовує зібрані дані для аналізу поведінкових патернів кожного користувача. Алгоритм визначає нормальну поведінку користувача, враховуючи його індивідуальні особливості (наприклад, звичні рухи очей, моргання, пози). Це дозволяє уникнути хибних спрацьовувань, які можуть виникати через індивідуальні особливості поведінки користувача.

– Класифікація поведінки за рівнем ризику.

На основі аналізу поведінкових патернів та критеріїв ризику система класифікує дії користувача на низький, середній або високий рівень ризику. Ця класифікація базується на інтенсивності, тривалості та частоті дій, а також на контексті, в якому вони відбуваються. Наприклад, одноразове відведення погляду від екрану може вважатися дією низького ризику, тоді як повторювана і тривала зміна позиції тіла або голови може бути класифікована як дія високого ризику.

– Ухвалення рішень на основі правил реагування.

Для кожного рівня ризику система застосовує набір правил, які визначають реакцію на конкретні дії.

Рішення при низькому ризику. Дії низького ризику фіксуються в журналі подій, але система не вживає жодних заходів. Це дозволяє зберегти повну історію поведінки користувача, не відволікаючи його від тестування.

Рішення при середньому ризику. Якщо система фіксує дії середнього ризику, користувач отримує автоматичне попередження, яке відображається на екрані. Це дає змогу користувачеві скоригувати свою поведінку без переривання тесту.

Рішення при високому ризику. При виявленні дій високого ризику система тимчасово блокує інтерфейс тестування та надсилає сповіщення адміністратору або проктору для негайного втручання.

Реакція системи в режимі реального часу:

Модель автоматизованого прийняття рішень забезпечує миттєву реакцію на підозрілі дії в режимі реального часу. Це дозволяє швидко реагувати на можливі порушення, мінімізуючи їхній вплив на процес тестування. Завдяки цьому система може:

– автоматично призупинити тестування. При виявленні серії дій високого ризику система може призупинити тестування і заблокувати інтерфейс, щоб уникнути подальших порушень;

– надіслати попередження та сповіщення. При середньому рівні ризику система надсилає попередження користувачеві, а у разі високого ризику – сповіщення адміністратору;

– журналювання подій для подальшого аналізу.

Кожна дія, що класифікується як середнього або високого рівня ризику, автоматично записується у журналі подій. Це дозволяє адміністраторам переглядати записи для подальшого аналізу і мати повний доступ до історії дій користувача під час тестування. Журналювання допомагає забезпечити об'єктивність і надає дані для потенційного перегляду результатів.

Розробимо алгоритм на основі даної інформації (рис.2.2).

Рисунок 2.2 – Модель алгоритму автоматизованого прийняття рішень на основі аналізу поведінки користувача

Розглянемо покроково даний алгоритм:

Вхідні дані:

- дія користувача: сукупність даних, що описують конкретну дію (наприклад, відведення погляду від екрану);

- еталонні поведінкові параметри: допустимі межі нормальної поведінки (базові дані);

- критерії оцінки: інтенсивність, тривалість, частота, контекст.

Вихідні дані:

- рівень ризику: низький, середній або високий.

Кроки алгоритму.

Крок 1: Оцінка інтенсивності дії.

Перевірка інтенсивності дії:

- визначити, наскільки значне відхилення дії від еталонного параметру;

- якщо інтенсивність відхилення незначна (наприклад, легке відведення погляду на декілька градусів), дія автоматично класифікується як низький ризик, і алгоритм переходить до наступної дії;

- якщо інтенсивність значна (наприклад, відведення погляду на великий кут або рух голови в сторону), перейти до кроку 2.

Крок 2: Оцінка тривалості дії.

Перевірка тривалості дії:

- виміряти, скільки часу триває дія;

- якщо дія триває короткий час (наприклад, кілька секунд), класифікувати її як низький ризик;

- якщо тривалість перевищує встановлений поріг, перейти до кроку 3.

Крок 3: Оцінка частоти дії.

Перевірка частоти повторення дії:

- визначити, як часто ця дія повторюється протягом певного періоду;

- якщо дія виконана одноразово або рідко повторюється, залишити рівень ризику як середній і перейти до наступної дії;

- якщо дія повторюється регулярно (наприклад, кожні кілька хвилин або секунд), перейти до кроку 4.

Крок 4: Оцінка контексту дії.

Аналіз контексту дії:

- оцінити, коли і за яких обставин була виконана дія (наприклад, під час відповіді на складне питання чи в момент паузи);

- якщо дія відбувається у відповідний контекстуально момент (наприклад, пауза), залишити рівень ризику як середній;

- якщо дія відбувається у контексті, який вказує на можливість порушення (наприклад, в момент відповіді на важке питання), присвоїти дії високий ризик.

Крок 5: Фінальна класифікація та присвоєння рівня ризику.

Визначення рівня ризику:

- низький ризик: дія має низьку інтенсивність, коротку тривалість, рідко повторюється і не має підозрілого контексту;

- середній ризик: дія має значну інтенсивність або тривалість, але контекст вказує на допустимість, або ж вона не має високої частоти;

- високий ризик: дія має високу інтенсивність, значну тривалість, повторюється з високою частотою та має контекст, який вказує на можливість порушення (наприклад, відведення погляду або рух голови в момент відповіді на складне питання).

Крок 6: Вихідне значення

Повернути рівень ризику дії:

- після перебору всіх критеріїв система повертає остаточне значення рівня ризику для поточної дії.

Приклад роботи алгоритму:

Припустимо, система зафіксувала, що користувач відвів погляд вбік:

- Інтенсивність. Відведення погляду значне (більше 30°), перейти до Кроку 2.

- Тривалість. Дія триває 5 секунд (перевищує поріг у 3 секунди), перейти до Кроку 3.

- Частота. Відведення погляду повторюється двічі протягом хвилини, перейти до Кроку 4.

- Контекст. Дія відбувається під час складного питання, що вказує на можливість порушення.

- Результат. Присвоїти дії високий ризик і надіслати сповіщення адміністратору.

Цей алгоритм ітеративно перебирає критерії та дозволяє системі точно визначати рівень ризику для кожної дії, аналізуючи інтенсивність, тривалість, частоту і контекст виконання дії.

2.5 Висновки до розділу

У другому розділі було здійснено проектування вдосконаленої системи оцінювання знань, що інтегрує модуль контролювання дій користувача з використанням технологій відстежування рухів і погляду. Основна мета розробки полягала у створенні механізму автоматизованого прийняття рішень, здатного забезпечити чесність та прозорість у процесі оцінювання знань, особливо в умовах дистанційного тестування.

Проаналізовані результати продемонстрували важливість адаптації системи контролю до індивідуальних особливостей користувачів, що дозволяє мінімізувати хибні спрацьовування та враховувати природні поведінкові особливості кожного користувача. Інтеграція технологій motion-tracking і eye-tracking дозволяє фіксувати навіть незначні відхилення від нормальної поведінки, такі як відведення погляду або нахили голови, що значно підвищує можливості виявлення потенційних спроб звернення до сторонніх матеріалів у режимі реального часу.

На основі цих технологій було розроблено алгоритм автоматизованого прийняття рішень, який аналізує поведінкові дані за кількома критеріями (інтенсивність, тривалість, частота, контекст) та класифікує дії користувача за рівнем ризику. Кожному рівню ризику відповідає певна реакція системи – від

журналювання подій до блокування тестування й надсилання сповіщень адміністратору для негайного втручання.

Система забезпечує адаптивне навчання моделі, що дозволяє алгоритмам вдосконалюватися на основі накопичених даних про поведінку користувачів, поступово підвищуючи точність і ефективність у наступних сесіях тестування. Завдяки автоматизації прийняття рішень зменшується суб’єктивний вплив прокторів, що підвищує об’єктивність і надійність результатів.

Таким чином, впровадження вдосконаленої системи оцінювання знань із автоматизованим контролем дій користувача створює об’єктивне і прозоре середовище для тестування, що значно знижує ризик шахрайства і сприяє чесному оцінюванню знань.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ВДОСКОНАЛЕНОЇ СИСТЕМИ ОЦІНЮВАННЯ ЗНАНЬ

У даному розділі буде описано програмну реалізацію вдосконаленої системи оцінювання знань, яка базується на інтеграції моделі контролювання дій користувача із технологіями відслідковування рухів і погляду. Основна увага приділяється вибору інструментів розробки, реалізації ключових модулів системи та їх інтеграції в єдину платформу.

Розділ містить обґрунтування вибору мови програмування та середовища розробки, детальну реалізацію модулів для оцінювання знань, відслідковування рухів і погляду користувача, а також створення графічного інтерфейсу, що забезпечує зручність роботи з системою. Крім того, проведено тестування реалізованої системи для перевірки її коректності та ефективності, а у висновках підведено підсумки щодо програмної реалізації та її відповідності поставленим задачам.

3.1 Обґрунтування вибору мови програмування

Для реалізації вдосконаленої системи оцінювання знань з інтеграцією модулів відстежування рухів і погляду користувача було обрано мову програмування Python (рис. 3.1). Вибір обґрунтовано її перевагами у виконанні завдань, які вимагають роботи з великими обсягами даних, інтеграції з технологіями комп’ютерного зору та машинного навчання, а також її широкою популярністю в розробці подібних систем. Розглянемо основні аспекти, які вплинули на цей вибір.

Рисунок 3.1 - Python

Python є мовою з простим синтаксисом, що дозволяє швидко розробляти програми, особливо в проєктах з великим обсягом експериментів і налаштувань. Це значно прискорює процес розробки та тестування модулів, таких як системи контролю дій користувача, які потребують частих змін і налаштувань для адаптації до реальних умов.

Python має розвинену екосистему бібліотек, що є однією з ключових причин вибору цієї мови програмування для розробки вдосконаленої системи оцінювання знань. Вона пропонує широкий набір інструментів, які значно спрощують реалізацію складних функціональних можливостей, необхідних для інтеграції модулів відстежування рухів і погляду користувача, а також для аналізу отриманих даних. Для обробки великих обсягів даних та їх аналізу бібліотеки NumPy та Pandas забезпечують ефективні інструменти для роботи з багатовимірними масивами, маніпуляції табличними даними та формування структурованих звітів. Їх використання дозволяє легко зберігати й обробляти дані, отримані від модулів контролю дій користувача, а також забезпечувати аналіз патернів поведінки [31].

Для реалізації функцій комп’ютерного зору та відстежування рухів і погляду ключовими є бібліотеки OpenCV, Mediapipe та Dlib. OpenCV надає можливості для аналізу зображень і відео, зокрема відстежування рухів тіла, голови чи рук, а також розпізнавання обличчя. Mediapipe, створена Google, дозволяє інтегрувати складні моделі відстежування рухів, таких як позиції тіла чи рис обличчя, що є критично важливим для системи контролю користувача. Dlib доповнює ці функції, забезпечуючи точне розпізнавання рис обличчя та очей навіть за умов змін освітлення. Для спеціалізованого відстежування погляду (eye-tracking) використовуються бібліотеки Pygaze та Pupil Labs SDK, які дозволяють зчитувати напрямок погляду, тривалість фокусування та аналізувати зміну уваги.

У сфері машинного навчання Python пропонує бібліотеки TensorFlow, PyTorch та Scikit-learn, які забезпечують створення та навчання моделей класифікації дій користувача за рівнями ризику. Ці бібліотеки дозволяють інтегрувати алгоритми адаптивного навчання, що поступово вдосконалюють систему, роблячи її точнішою та ефективнішою з кожною наступною сесією

тестування. Для обробки сигналів і взаємодії з апаратними засобами Python пропонує бібліотеки PySerial і PyGame, які дозволяють працювати з сенсорами рухів та іншими пристроями, а також створювати інтерактивні візуалізації для моніторингу поведінки користувачів у реальному часі [32].

Для створення інтерфейсу користувача застосовуються такі інструменти, як Tkinter для настільних застосунків або Dash для веб-додатків, що забезпечують зручний і функціональний доступ до системи. Завдяки розвиненій екосистемі бібліотек Python не лише спрощує реалізацію складних функцій, а й забезпечує високу ефективність роботи, гнучкість у створенні інтегрованих рішень, а також підтримує швидке прототипування і тестування. Така екосистема робить Python ідеальним вибором для розробки інтегрованої системи оцінювання знань із підтримкою технологій відстежування рухів і погляду.

Python забезпечує високу продуктивність у виконанні дослідницьких і аналітичних завдань завдяки своїй гнучкості, широкій екосистемі бібліотек і можливості інтеграції з високопродуктивними мовами програмування. У контексті розробки вдосконаленої системи оцінювання знань ця продуктивність дозволяє швидко прототипувати, тестувати та вдосконалювати алгоритми, які аналізують поведінкові дані користувача, відстежують рухи та погляд і класифікують дії за рівнем ризику [33].

Одна з основних причин високої продуктивності Python полягає у підтримці бібліотек, спеціалізованих на числових обчисленнях і обробці великих обсягів даних. Наприклад, бібліотека NumPy дозволяє ефективно виконувати операції з багатовимірними масивами, які використовуються для аналізу координат рухів або траєкторій погляду користувача. Завдяки оптимізації на основі C-імплементацій, такі обчислення виконуються швидко навіть на великих наборах даних. Pandas, у свою чергу, пропонує інструменти для обробки структурованих даних, дозволяючи швидко фільтрувати, групувати та аналізувати інформацію, отриману від модулів контролю.

Python також забезпечує легкий доступ до потужних бібліотек для роботи з комп’ютерним зором і машинним навчанням, таких як OpenCV, TensorFlow і

PyTorch. Ці інструменти оптимізовані для високої продуктивності завдяки підтримці багатопоточності та апаратного прискорення, зокрема через використання графічних процесорів (GPU). Це особливо важливо для аналізу відеопотоків у реальному часі, де система повинна швидко обробляти велику кількість кадрів для відстежування рухів і визначення підозрілої поведінки. Для машинного навчання Python дозволяє швидко розробляти, навчати та вдосконалювати моделі класифікації, використовуючи оптимізовані функції для обчислення градієнтів, побудови нейронних мереж і виконання матричних операцій [34].

Ще однією перевагою Python є можливість інтеграції з іншими високопродуктивними мовами, такими як C і C++. За допомогою Cython або бібліотеки ctypes розробник може перенести найкритичніші для продуктивності частини програми на C або C++, зберігаючи простоту основної логіки на Python. Це дозволяє досягти оптимального співвідношення між швидкістю виконання і зручністю розробки.

Python також підтримує інтерактивне тестування та аналіз, наприклад, у середовищах Jupyter Notebook. Це забезпечує можливість швидко візуалізувати результати експериментів, налаштовувати параметри алгоритмів і одразу бачити їхній вплив на точність чи швидкість роботи системи. У проєкті, де важливе точне й швидке розпізнавання дій користувача, така інтерактивність значно прискорює цикл розробки.

Завдяки своїй високій продуктивності Python забезпечує швидке виконання дослідницьких і аналітичних завдань, необхідних для розробки вдосконаленої системи оцінювання знань. Він дозволяє оптимально поєднувати швидкість обчислень, багатофункціональність бібліотек і зручність розробки, забезпечуючи надійну платформу для створення інноваційних систем аналізу та контролю.

Python був обраний для реалізації вдосконаленої системи оцінювання знань завдяки своїм унікальним перевагам, які виявляються особливо важливими в контексті завдань відстежування рухів і погляду користувача, обробки поведінкових даних, машинного навчання та інтеграції різних компонентів

системи. Порівняння Python з іншими популярними мовами програмування, такими як Java, C++ та JavaScript, підтверджує його оптимальність для цього проєкту [34].

Python vs Java

Java є потужною мовою, яка забезпечує високу продуктивність і надійність у багатопотокових програмах, але має кілька недоліків у контексті поставлених завдань. Реалізація алгоритмів комп'ютерного зору або машинного навчання у Java вимагає більше зусиль, оскільки її екосистема бібліотек значно поступається Python. Наприклад, бібліотеки OpenCV і TensorFlow, хоча й доступні для Java, мають ширший функціонал і кращу інтеграцію в Python. Крім того, синтаксис Java є складнішим, що ускладнює швидке прототипування і тестування системи. Python, навпаки, забезпечує простий і зрозумілий синтаксис, що дозволяє зосередитися на розв'язанні задач, а не на написанні коду.

Python vs C++

C++ відомий своєю високою продуктивністю і прямим доступом до апаратних ресурсів, що може бути корисним для програм, які вимагають обчислювальної потужності. Проте, у випадку проєкту, що передбачає часте оновлення та адаптацію алгоритмів, C++ виявляється менш зручним. Його складний синтаксис і триваліший процес розробки значно уповільнюють створення прототипів і реалізацію модулів. Крім того, робота з бібліотеками для машинного навчання (наприклад, TensorFlow або PyTorch) у C++ є складнішою і менш зручною. Python, своєю чергою, надає інтуїтивно зрозумілий інтерфейс до цих бібліотек, що дозволяє розробляти складні моделі з мінімальними витратами часу [35].

Python vs JavaScript

JavaScript є чудовим вибором для створення інтерфейсів користувача, особливо у веб-додатках. Однак, його можливості в обробці даних і виконанні завдань машинного навчання значно обмежені у порівнянні з Python. Хоча для JavaScript доступні фреймворки, такі як TensorFlow.js, вони не забезпечують тієї ж продуктивності й гнучкості, які пропонує Python. Також JavaScript не має такої

багатої екосистеми бібліотек для комп’ютерного зору, як OpenCV чи Mediapipe, що є критично важливими для реалізації функцій motion-tracking і eye-tracking. Python значно переважає у завданнях обробки великих обсягів даних і створення адаптивних алгоритмів.

Python vs спеціалізовані мови (наприклад, R)

R є популярною мовою для статистичного аналізу і роботи з даними, але її можливості в комп’ютерному зорі та машинному навчанні обмежені. Крім того, R не надає такої ж підтримки багатофункціональних бібліотек для інтеграції з апаратними засобами, як Python. У випадку створення багатокомпонентної системи Python є універсальнішим вибором, дозволяючи об’єднати статистичний аналіз, машинне навчання і взаємодію з апаратними засобами в одному середовищі.

Python поєднує простоту синтаксису, універсальність і розвинену екосистему бібліотек, таких як NumPy, Pandas, OpenCV, TensorFlow, Mediapipe та інші. Це дозволяє швидко прототипувати, тестувати і впроваджувати складні системи з мінімальними витратами часу. Крім того, активна спільнота розробників і велика кількість навчальних ресурсів роблять Python доступним для розробників з різним рівнем досвіду [35].

Порівняння Python з іншими мовами програмування показало, що він є оптимальним вибором для реалізації вдосконаленої системи оцінювання знань. Його переваги включають простоту, потужну екосистему бібліотек для комп’ютерного зору, машинного навчання та обробки даних, а також здатність інтегрувати різні компоненти системи. Це дозволяє досягти високої продуктивності, гнучкості і швидкості розробки, що є критично важливими для створення ефективної і надійної системи.

3.2 Обґрунтування вибору середовища розробки

Для реалізації вдосконаленої системи оцінювання знань було обрано середовище розробки PyCharm (рис. 3.2), розроблене компанією JetBrains. Це інтегроване середовище розробки (IDE) створене спеціально для Python і має

широкий спектр інструментів, які значно спрощують написання, тестування, налагодження та управління великими проєктами. У розробці системи з використанням Python, яка інтегрує технології відстежування рухів і погляду користувача, PyCharm забезпечує зручність і ефективність на кожному етапі роботи [36].

PyCharm забезпечує автоматичне підсвічування синтаксису Python, що дозволяє розробникам легко орієнтуватися в коді. Це особливо важливо у великих проєктах, де кожен модуль виконує складні функції. Наприклад, у модулі аналізу поведінки користувача можуть бути десятки функцій для обробки даних із сенсорів, і підсвічування синтаксису допомагає швидко ідентифікувати помилки чи важливі частини коду. Крім того, PyCharm пропонує автозаповнення коду (code completion), яке підказує доступні методи, функції та змінні, що значно пришвидшує написання програм і знижує ймовірність помилок.

Рисунок 3.2 – PyCharm

Однією з ключових переваг PyCharm є його потужний вбудований дебаггер, який дозволяє покроково виконувати код, аналізувати стан змінних і відстежувати виконання програми. Для системи оцінювання знань, яка аналізує великі обсяги даних у реальному часі, дебаггер є критично важливим. Наприклад, під час роботи модулів motion-tracking та eye-tracking можуть виникати складні помилки, пов’язані з обробкою відеопотоків або аналізом координат рухів. Зручний дебаггер PyCharm дозволяє швидко локалізувати проблеми та знаходити їхнє вирішення, забезпечуючи безперебійну роботу системи [37].

PyCharm забезпечує повну інтеграцію з усіма основними бібліотеками Python, які використовуються в розробці вдосконаленої системи оцінювання знань. Наприклад:

- OpenCV та Mediapipe для обробки відеопотоків і реалізації функцій motion-tracking та eye-tracking;

- TensorFlow та PyTorch для створення та навчання моделей машинного навчання, які використовуються для класифікації поведінки користувача за рівнем ризику;

- NumPy та Pandas для обробки даних, отриманих із сенсорів і камер. PyCharm дозволяє легко встановлювати та оновлювати ці бібліотеки через інтегровані менеджери пакетів (pip, conda), що значно спрощує управління залежностями проєкту [38].

PyCharm надає можливість створювати ізольовані віртуальні середовища (virtual environments) для кожного проєкту. У проєкті, де є кілька модулів із різними наборами залежностей, ця функція є особливо важливою. Наприклад, модуль motion-tracking може потребувати однієї версії OpenCV, тоді як модуль eye-tracking – іншої. Використання віртуальних середовищ дозволяє уникнути конфліктів між залежностями та забезпечує стабільність системи.

PyCharm має вбудовані засоби для роботи з базами даних, що є надзвичайно корисним для зберігання результатів тестування, логів поведінки користувачів та іншої інформації. Підтримка SQL-запитів прямо з IDE дозволяє розробникам ефективно управляти даними, створювати звіти та проводити аналіз.

PyCharm інтегрується з популярними фреймворками для тестування, такими як pytest та unittest, що дозволяє створювати автоматизовані тести для кожного модуля системи. Наприклад, можна перевірити, чи правильно класифікується поведінка користувача, чи коректно працює алгоритм відстежування рухів. Завдяки цьому забезпечується висока якість системи та знижується ризик помилок [39].

PyCharm має вбудовану підтримку для Git та інших систем контролю версій. Це дозволяє ефективно управляти змінами в коді, створювати гілки для різних

функціональних модулів і зберігати історію змін. У командних проєктах ця функція значно полегшує співпрацю між розробниками.

PyCharm працює на основних операційних системах, таких як Windows, macOS і Linux, що забезпечує універсальність середовища розробки. Це дозволяє розробникам використовувати PyCharm незалежно від операційної системи, а також тестувати систему на різних платформах, забезпечуючи її сумісність.

PyCharm забезпечує просту інтеграцію з TensorFlow, PyTorch та іншими фреймворками, що спрощує процес навчання моделей для класифікації дій користувача. У системі оцінювання знань це дозволяє ефективно створювати та тестувати моделі, які адаптивно визначають рівень ризику поведінки.

PyCharm має активну спільноту користувачів і розробників, а також велику базу навчальних матеріалів. Це дозволяє швидко знаходити відповіді на запитання, які можуть виникнути під час роботи над проєктом, та вирішувати технічні проблеми [40].

PyCharm є ідеальним середовищем розробки для проєктів, побудованих на Python, завдяки своїм потужним інструментам для написання, тестування та налагодження коду. Воно забезпечує легку інтеграцію з бібліотеками для комп’ютерного зору, машинного навчання та обробки даних, дозволяє ефективно працювати з базами даних і забезпечує стабільне управління залежностями. Завдяки PyCharm розробка вдосконаленої системи оцінювання знань стає швидкою, ефективною і зручною, що гарантує високу якість і надійність кінцевого продукту.

3.3 Реалізація модуля система оцінювання знань

Реалізація модуля системи оцінювання знань є одним із ключових етапів розробки проєкту, який забезпечує виконання основної функції — проведення тестування користувачів із застосуванням вдосконаленої моделі контролювання їхніх дій. Цей модуль інтегрується з іншими компонентами системи, такими як модулі відстежування рухів і погляду користувача, та включає кілька

функціональних підсистем, які забезпечують гнучкість, адаптивність і захист від шахрайства під час оцінювання знань.

Для зберігання питань, варіантів відповідей та результатів тестів ми підключаємо базу даних, використовуючи бібліотеку SQLAlchemy. Наступний код реалізує підключення:

```
from flask_sqlalchemy import SQLAlchemy
db = SQLAlchemy()
```

Моделюємо структуру бази даних для зберігання тексту питань і варіантів відповідей. Реалізуємо це за допомогою наступного коду:

```
class Question(db.Model):
    id = db.Column(db.Integer, primary_key=True)  # Ідентифікатор питання
    text = db.Column(db.String(255), nullable=False)  # Текст питання
    options = db.relationship('Option', backref='question', lazy=True)  # Варіанти відповідей
    correct_option_id = db.Column(db.Integer, nullable=False)  # Правильна відповідь

class Option(db.Model):
    id = db.Column(db.Integer, primary_key=True)  # Ідентифікатор варіанту
    question_id = db.Column(db.Integer, db.ForeignKey('question.id'), nullable=False)  # Посилання на питання
    text = db.Column(db.String(255), nullable=False)  # Текст варіанту
```

Наступним кроком реалізуємо функцію, яка дозволяє адміністраторам створювати нові питання разом із варіантами відповідей. Код нижче відповідає за прийняття даних у форматі JSON і запис їх у базу даних:

```
@app.route('/create-question', methods=['POST'])
def create_question():
    data = request.json  # Отримуємо дані від клієнта
    question = Question(text=data['text'], correct_option_id=data['correct_option_id'])
    db.session.add(question)
    db.session.commit()
    # Додаємо варіанти відповідей
    for option_text in data['options']:
        option = Option(question_id=question.id, text=option_text)
        db.session.add(option)
    db.session.commit()
    return jsonify({"message": "Question created"}), 201
```

Для відображення питань користувачеві створюємо функцію, яка отримує всі питання з бази даних разом із їхніми варіантами відповідей:

```
@app.route('/get-questions', methods=['GET'])
def get_questions():
    questions = Question.query.all()  # Отримуємо всі питання
    result = []
    for question in questions:
        options = Option.query.filter_by(question_id=question.id).all()  # Отримуємо варіанти
        result.append({
            "id": question.id,
            "text": question.text,
            "options": [{"id": o.id, "text": o.text} for o in options],
```

```
            "correct_option_id": question.correct_option_id
        })
    return jsonify(result), 200
```

Для перевірки правильності відповідей користувача реалізуємо ендпоінт, який приймає відповіді у форматі JSON, порівнює їх із правильними і повертає підсумковий результат:

```
@app.route('/start-test', methods=['POST'])
def start_test():
    data = request.json  # Отримуємо відповіді користувача
    user_answers = data['answers']  # Словник {question_id: selected_option_id}
    score = 0

    for question_id, selected_option_id in user_answers.items():
        question = Question.query.get(question_id)  # Отримуємо питання
        if question.correct_option_id == selected_option_id:  # Перевіряємо правильність відповіді
            score += 1

    return jsonify({"message": "Test completed", "score": score}), 200
```

Для підвищення безпеки тестування додаємо функцію, яка обробляє поведінкові дані користувача, такі як напрямок погляду або рухи тіла, і повертає попередження про підозрілі дії:

```
@app.route('/track-user-behavior', methods=['POST'])
def track_user_behavior():
    behavior_data = request.json  # Дані від motion-tracking чи eye-tracking модулів
    # Виявлення підозрілої поведінки
    if behavior_data['gaze_direction'][0] > 50:  # Значне відведення погляду
        return jsonify({"warning": "Suspicious activity detected"}), 200
    return jsonify({"message": "Behavior normal"}), 200
```

Для забезпечення взаємодії користувача із системою реалізуємо простий HTML+JavaScript інтерфейс, який отримує питання через API, дозволяє користувачеві відповідати і надсилає результати для перевірки:

```
<!DOCTYPE html>
<html>
<head>
    <title>Knowledge Test</title>
</head>
<body>
    <h1>Test</h1>
    <form id="test-form">
        <div id="questions-container"></div>
        <button type="submit">Submit</button>
    </form>

    <script>
        // Отримуємо питання з сервера
        fetch('/get-questions')
            .then(response => response.json())
            .then(data => {
```

```
            const container = document.getElementById('questions-container');
            data.forEach(question => {
              const questionDiv = document.createElement('div');
              questionDiv.innerHTML = `<p>${question.text}</p>`;
              question.options.forEach(option => {
                questionDiv.innerHTML += `
                  <input type="radio" name="question-${question.id}" value="${option.id}">
                  ${option.text}<br>`;
              });
              container.appendChild(questionDiv);
            });
          });

        // Надсилаємо відповіді користувача на сервер
        document.getElementById('test-form').addEventListener('submit', (e) => {
          e.preventDefault();
          const answers = {};
          const inputs = document.querySelectorAll('input[type="radio"]:checked');
          inputs.forEach(input => {
            const questionId = input.name.split('-')[1];
            answers[questionId] = parseInt(input.value);
          });

          fetch('/start-test', {
            method: 'POST',
            headers: { 'Content-Type': 'application/json' },
            body: JSON.stringify({ answers })
          })
          .then(response => response.json())
          .then(data => alert(`Your score: ${data.score}`));
        });
      </script>
    </body>
    </html>
```

Ця покрокова реалізація забезпечує створення повноцінної системи тестування. Було налаштовано базу даних, реалізовано функції для керування питаннями та обробки відповідей, інтегровано модулі контролю поведінки і створено простий інтерфейс для користувача.

3.4 Реалізаця модуля відслідковування рухів користувача

Модуль відслідковування рухів користувача реалізується на основі бібліотеки OpenCV (або Mediapipe) для обробки відеопотоків у реальному часі. Цей модуль аналізує рухи голови, рук або тіла користувача та інтегрується із системою оцінювання знань для виявлення підозрілої поведінки, яка може свідчити про шахрайство.

Для роботи з відеопотоками, аналізу рухів і відображення результатів імпортуємо бібліотеки OpenCV та Mediapipe:

```
import cv2
import mediapipe as mp
```

Використовуємо Mediapipe для детекції та відстежування позицій частин тіла:

```
mp_pose = mp.solutions.pose  # Модуль для аналізу рухів тіла
pose = mp_pose.Pose()  # Ініціалізація моделі Mediapipe для відстежування
mp_drawing = mp.solutions.drawing_utils  # Інструмент для візуалізації
```

Реалізуємо захоплення відеопотоку з вебкамери, який буде оброблятися в реальному часі:

```
cap = cv2.VideoCapture(0)  # Використання вебкамери

if not cap.isOpened():
   print("Error: Cannot access the camera")
   exit()
```

На кожному кадрі визначаємо ключові точки тіла (наприклад, положення голови, плечей, рук) та аналізуємо їхні зміни:

```
while cap.isOpened():
   ret, frame = cap.read()
   if not ret:
      break
   # Зміна формату кадру для Mediapipe
   frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
   results = pose.process(frame_rgb)
   # Відображення ключових точок і зв'язків
   if results.pose_landmarks:
      mp_drawing.draw_landmarks(frame, results.pose_landmarks,
mp_pose.POSE_CONNECTIONS)

      # Отримання координат голови (напр. точка 0 - ніс)
      nose = results.pose_landmarks.landmark[mp_pose.PoseLandmark.NOSE]
      print(f"Голова користувача: X={nose.x}, Y={nose.y}, Z={nose.z}")

   # Відображення кадру
   cv2.imshow('Motion Tracking', frame)

   # Вихід за натисканням клавіші "q"
   if cv2.waitKey(10) & 0xFF == ord('q'):
      break
```

Додамо аналіз рухів голови чи рук для виявлення відхилень від нормальної поведінки:

```
# Простий приклад виявлення підозрілої поведінки
if nose.x < 0.3 or nose.x > 0.7:  # Голова користувача сильно нахилена вліво чи вправо
   print("Підозріла поведінка: сильний нахил голови")
```

Результати відстежування передаються до сервера для подальшої обробки. Реалізуємо відправку цих даних через POST-запит:

```
import requests

data = {
   "gaze_direction": [nose.x, nose.y, nose.z],
   "head_movement": "tilt" if nose.x < 0.3 or nose.x > 0.7 else "normal"
}

response = requests.post("http://localhost:5000/track-user-behavior", json=data)
print(response.json())
```

Реалізований модуль відслідковування рухів користувача здатний аналізувати відеопотоки в реальному часі, визначати ключові точки тіла та виявляти підозрілу поведінку, таку як нахили голови чи відхилення від екрану. Завдяки інтеграції з основною системою через API цей модуль ефективно доповнює функціонал системи оцінювання знань.

3.5 Реалізація модуля відслідковування погляду користувача

Модуль відслідковування погляду (eye-tracking) відповідає за аналіз напрямку погляду користувача під час тестування. Основна мета — виявлення відхилень погляду від екрану або інших підозрілих дій, таких як часте переведення погляду в сторони. Для реалізації використовується бібліотека OpenCV разом із моделями, які розпізнають очі та визначають напрямок погляду.

Для обробки відеопотоків та виявлення обличчя та очей використовуємо OpenCV:

```
face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
eye_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_eye.xml')
```

Захоплюємо відеопотік із вебкамери:

```
cap = cv2.VideoCapture(0)

if not cap.isOpened():
   print("Error: Cannot access the camera")
   exit()
```

На кожному кадрі виконується розпізнавання обличчя та очей:

```
while cap.isOpened():
   ret, frame = cap.read()
   if not ret:
      break

   # Перетворення кадру в чорно-білий формат для підвищення точності
   gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

   # Виявлення обличчя
```

```
faces = face_cascade.detectMultiScale(gray, scaleFactor=1.3, minNeighbors=5)
for (x, y, w, h) in faces:
    # Відображення обличчя
    cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 0, 0), 2)

    # Виявлення очей усередині області обличчя
    roi_gray = gray[y:y + h, x:x + w]
    roi_color = frame[y:y + h, x:x + w]
    eyes = eye_cascade.detectMultiScale(roi_gray)
    for (ex, ey, ew, eh) in eyes:
        cv2.rectangle(roi_color, (ex, ey), (ex + ew, ey + eh), (0, 255, 0), 2)
```

Для визначення напрямку погляду аналізуємо положення очей у межах виявленого обличчя:

```
for (ex, ey, ew, eh) in eyes:
    eye_center_x = ex + ew // 2  # Центр ока
    face_center_x = w // 2  # Центр обличчя

    # Визначення напрямку погляду
    if eye_center_x < face_center_x - 20:  # Погляд ліворуч
        print("Погляд ліворуч")
    elif eye_center_x > face_center_x + 20:  # Погляд праворуч
        print("Погляд праворуч")
    else:
        print("Погляд прямо")
```

Результати відстежування передаються до сервера через API для фіксації підозрілих дій:

```
import requests

data = {
    "gaze_direction": "left" if eye_center_x < face_center_x - 20 else
                      "right" if eye_center_x > face_center_x + 20 else
                      "center"
}

response = requests.post("http://localhost:5000/track-user-behavior", json=data)
print(response.json())
```

Реалізований модуль відслідковування погляду користувача забезпечує аналіз напрямку погляду в реальному часі, фіксує підозрілі відхилення та передає ці дані до основної системи через API. Завдяки цьому досягається підвищення надійності оцінювання знань та запобігання шахрайству під час тестування.

3.6 Реалізаця графічного інтерфейсу користувача

Графічний інтерфейс користувача (GUI) є ключовим елементом системи оцінювання знань. Він дозволяє учасникам тестування легко взаємодіяти із системою, переглядати запитання, обирати відповіді, отримувати результати тесту,

а також забезпечує інтеграцію із модулями відстежування рухів і погляду. Для створення GUI можна використовувати Flask у поєднанні з HTML/CSS/JavaScript для веб-інтерфейсу або Tkinter для настільного застосунку.

Ініціалізуємо Flask-додаток, який буде обробляти запити клієнта та надавати йому веб-сторінки для тестування:

```
from flask import Flask, render_template, request, jsonify

app = Flask(__name__)

# Точка входу для головної сторінки
@app.route('/')
def index():
    return render_template('index.html')
```

Розробляємо HTML-сторінку index.html, яка містить базову структуру інтерфейсу:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Knowledge Test</title>
    <style>
        body { font-family: Arial, sans-serif; margin: 20px; }
        h1 { text-align: center; }
        .question { margin-bottom: 20px; }
        .option { margin-left: 20px; }
    </style>
</head>
<body>
    <h1>Knowledge Test</h1>
    <div id="questions-container"></div>
    <button id="submit-btn" style="display: none;">Submit</button>

    <script>
        // Завантаження питань із сервера
        fetch('/get-questions')
            .then(response => response.json())
            .then(data => {
                const container = document.getElementById('questions-container');
                data.forEach((question, index) => {
                    const questionDiv = document.createElement('div');
                    questionDiv.classList.add('question');
                    questionDiv.innerHTML = `<p><strong>${index + 1}. ${question.text}</strong></p>`;
                    question.options.forEach(option => {
                        questionDiv.innerHTML += `
                            <div class="option">
                                <input type="radio" name="question-${question.id}" value="${option.id}">
                                <label>${option.text}</label>
                            </div>`;
                    });
                    container.appendChild(questionDiv);
```

```
            });

            // Відображення кнопки "Submit", якщо є питання
            if (data.length > 0) {
               document.getElementById('submit-btn').style.display = 'block';
            }
         });

      // Надсилання відповідей користувача
      document.getElementById('submit-btn').addEventListener('click', () => {
         const answers = {};
         const inputs = document.querySelectorAll('input[type="radio"]:checked');
         inputs.forEach(input => {
            const questionId = input.name.split('-')[1];
            answers[questionId] = parseInt(input.value);
         });

         fetch('/start-test', {
            method: 'POST',
            headers: { 'Content-Type': 'application/json' },
            body: JSON.stringify({ answers })
         })
         .then(response => response.json())
         .then(data => alert(`Your score: ${data.score}`));
      });
   </script>
</body>
</html>
```

Додаємо серверну логіку для обробки запитів, які надають питання та обробляють відповіді:

```
@app.route('/get-questions', methods=['GET'])
def get_questions():
   # Приклад питань для тесту
   questions = [
      {
         "id": 1,
         "text": "What is the capital of France?",
         "options": [
            {"id": 1, "text": "Paris"},
            {"id": 2, "text": "London"},
            {"id": 3, "text": "Berlin"}
         ]
      },
      {
         "id": 2,
         "text": "What is 2 + 2?",
         "options": [
            {"id": 4, "text": "3"},
            {"id": 5, "text": "4"},
            {"id": 6, "text": "5"}
         ]
      }
   ]
   return jsonify(questions)

@app.route('/start-test', methods=['POST'])
```

```
def start_test():
    data = request.json  # Отримуємо відповіді користувача
    correct_answers = {1: 1, 2: 5}  # Правильні відповіді
    score = 0

    for question_id, selected_option in data['answers'].items():
        if correct_answers.get(int(question_id)) == selected_option:
            score += 1

    return jsonify({"message": "Test completed", "score": score})
```

Альтернативний варіант реалізації GUI – використання бібліотеки Tkinter для створення настільного застосунку.

```
import tkinter as tk
from tkinter import messagebox

# Питання для тесту
questions = [
    {"text": "What is the capital of France?", "options": ["Paris", "London", "Berlin"], "correct":
0},
    {"text": "What is 2 + 2?", "options": ["3", "4", "5"], "correct": 1}
]
class QuizApp:
    def __init__(self, root):
        self.root = root
        self.root.title("Knowledge Test")
        self.current_question = 0
        self.score = 0

        self.question_label = tk.Label(root, text="", font=("Arial", 16), wraplength=400)
        self.question_label.pack(pady=20)

        self.options = []
        for i in range(3):
            btn = tk.Radiobutton(root, text="", value=i, variable=tk.IntVar())
            btn.pack(anchor="w")
            self.options.append(btn)

        self.next_button = tk.Button(root, text="Next", command=self.next_question)
        self.next_button.pack(pady=20)

        self.show_question()

    def show_question(self):
        question = questions[self.current_question]
        self.question_label.config(text=question["text"])
        for i, option in enumerate(question["options"]):
            self.options[i].config(text=option)

    def next_question(self):
        selected_option = None
        for i, btn in enumerate(self.options):
            if btn.var.get() == 1:
                selected_option = i

        if selected_option == questions[self.current_question]["correct"]:
            self.score += 1
```

```
            self.current_question += 1
            if self.current_question < len(questions):
                self.show_question()
            else:
                messagebox.showinfo("Test Completed", f"Your score: {self.score}/{len(questions)}")
                self.root.quit()

root = tk.Tk()
app = QuizApp(root)
root.mainloop()
```

Графічний інтерфейс користувача забезпечує простоту та зручність проходження тестів. Реалізація через Flask дозволяє створювати адаптивний веб-інтерфейс, тоді як Tkinter підходить для настільного застосунку. Обидва підходи інтегруються з іншими модулями системи для забезпечення повного функціоналу.

3.7 Тестування реалізованої системи

Здійснимо тестування реалізованої системи, для початку протестуємо систему відслідковування рухів та погляду для цього виведемо показники нахилу обличчя кількості моргань та інші.

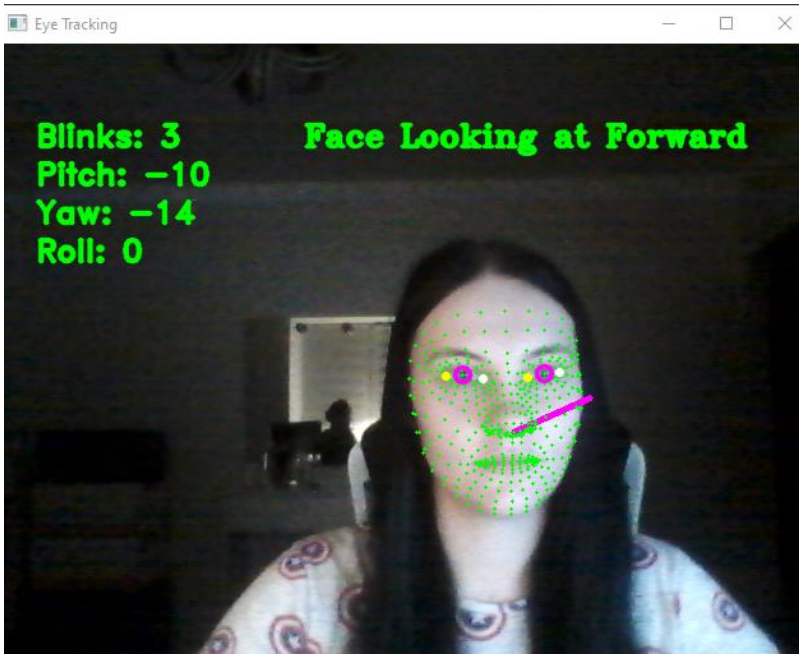


Рисунок 3.3 – Розпізнавання обличчя

Після зміни положення обличчя і декількох кліпань видно, що показники змінилися (рис.3.4).

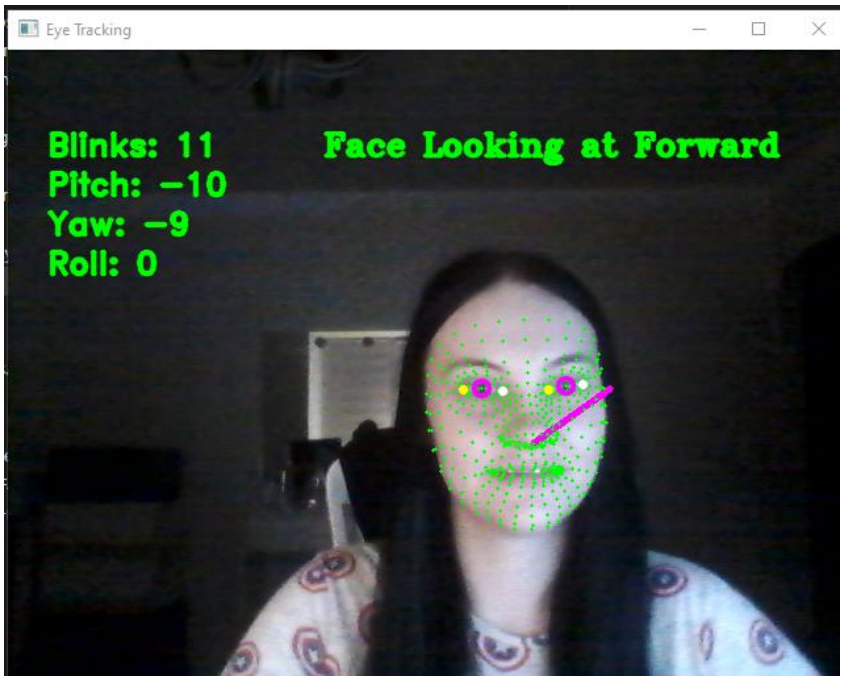


Рисунок 3.4 – Розпізнавання обличчя з змінинеми показниками

Змінимо положення обличчя вправо і перевіримо показники (рис.3.5).

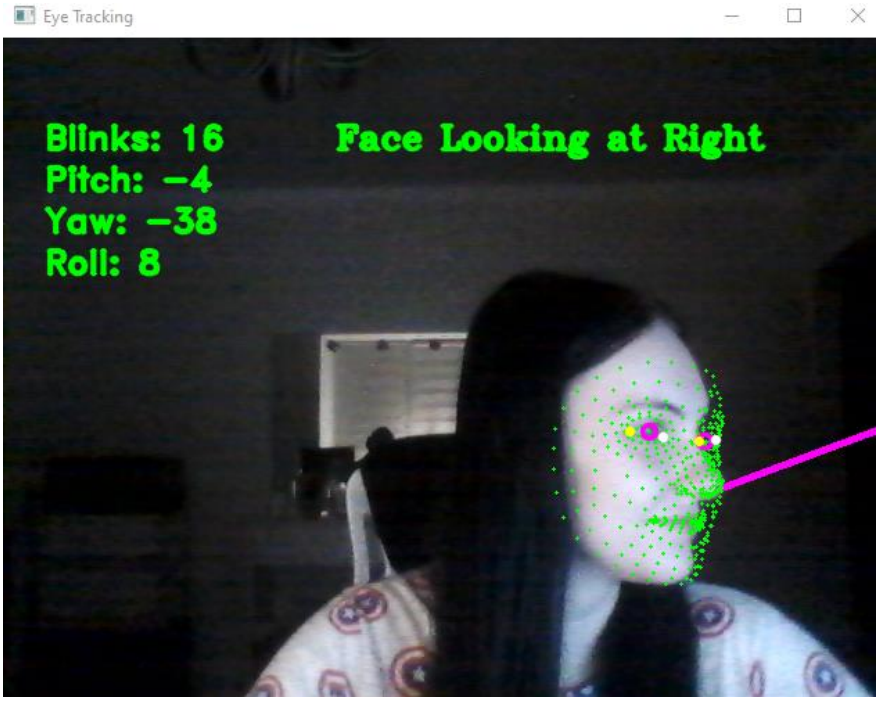


Рисунок 3.5 – Положення обличчя вправо

Як видно з показників, алгоритм зафіксував поворот обличчя, що є підставою для попередження користувача.

Протестуємо повернувши обличчя вліво (рис.3.6).

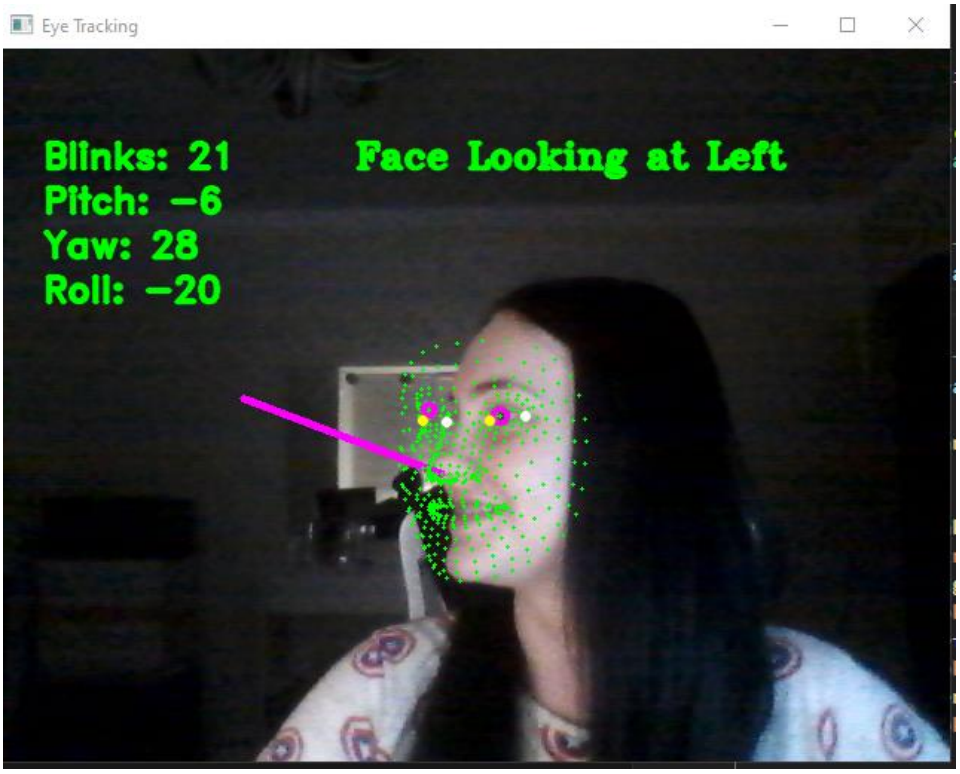


Рисунок 3.6 – Положення обличчя зліва

Як видно з показників, алгоритм зафіксував поворот обличчя, що є підставою для попередження користувача.

Далі протекстуємо розпізнавання з горизонтальними поворотами обличчя.

Протестуємо погляд вверх (рис.3.7).

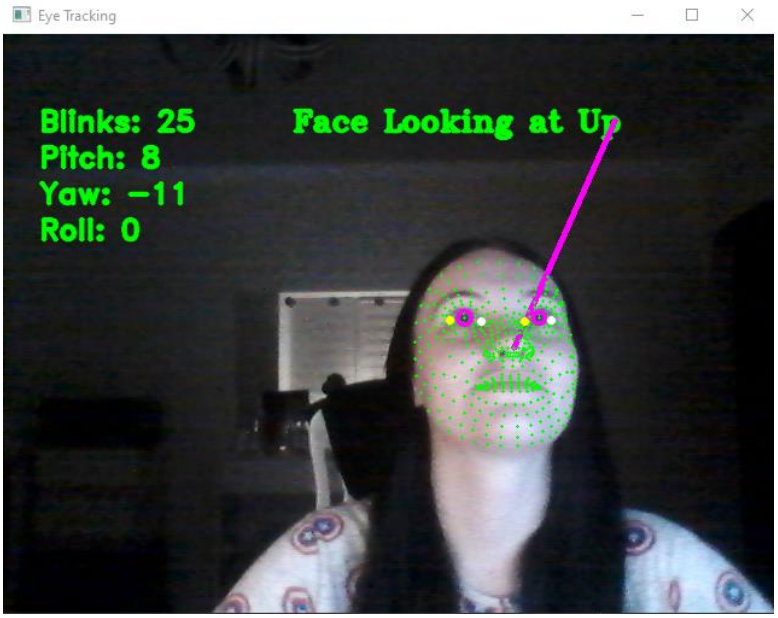


Рисунок 3.7 – Положення обличчя з поглядом вверх

На даному рисунку видно, що погляд був зафіксовий вгору, що є показником вірності роботи алгоритму.

Також потрібно протестувати найбільш важливий крок, це погляд вниз (рис.3.8).

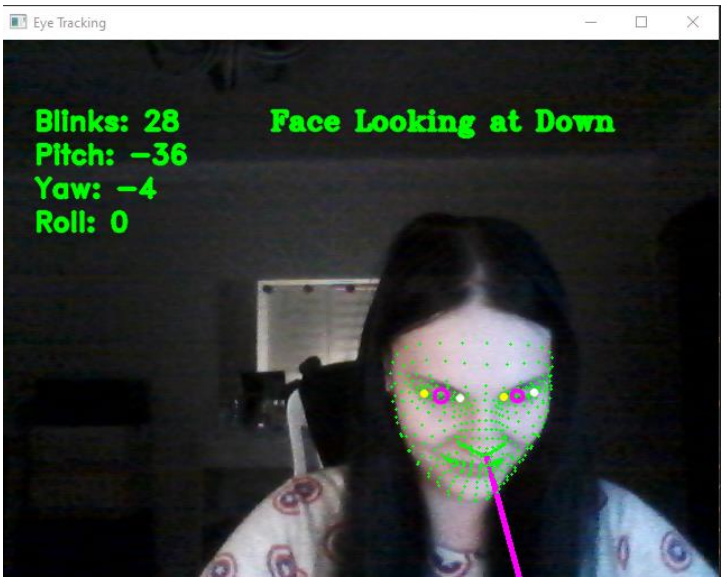


Рисунок 3.8 – Положення обличчя з поглядом вниз

На рисунку 3.7 чудово видно, що алгоритм зафіксував відведення погляду і зміну положення обличчя.

Також переглянемо логи які були введені, для впевненості вірності роботи алгоритму.

```
Head Pose Angles: Pitch=-2.1391682976739617, Yaw=6.242275815548322, Roll=1.2324855575906588
Sent UDP packet to ('127.0.0.1', 7070): b'\x01\x13\xe0_\x93\x01\x00\x00\x99\x01\x00\x00\x1c\x01\x00\x00N\x00\x00\x00\xfe\xff\xff\xff'
Total Blinks: 6
Left Eye Center X: 409.0 Y: 284.5
Right Eye Center X: 344.0 Y: 285.5
Left Iris Relative Pos Dx: 78 Dy: -2
Right Iris Relative Pos Dx: -49 Dy: -1

Head Pose Angles: Pitch=-2.252598170328426, Yaw=5.870480974060714, Roll=1.185591685770935
Sent UDP packet to ('127.0.0.1', 7070): b'Y\x13\xe0_\x93\x01\x00\x00\x99\x01\x00\x00\x1c\x01\x00\x00N\x00\x00\x00\xfe\xff\xff\xff'
Total Blinks: 6
Left Eye Center X: 409.0 Y: 284.5
Right Eye Center X: 344.0 Y: 285.5
Left Iris Relative Pos Dx: 78 Dy: -2
Right Iris Relative Pos Dx: -49 Dy: -1

Head Pose Angles: Pitch=-2.344588790497352, Yaw=5.490432183676855, Roll=1.2218295472993994
```

```
Sent UDP packet to ('127.0.0.1', 7070): b'\xa1\x13\xe0_\x93\x01\x00\x00\x99\x01\x00\x00\x1c\x01\x00\x00N\x00\x00\x00\xfe\xff\xff\xff'
Total Blinks: 6
Left Eye Center X: 408.5 Y: 284.5
Right Eye Center X: 344.0 Y: 285.5
Left Iris Relative Pos Dx: 77 Dy: -2
Right Iris Relative Pos Dx: -49 Dy: -1

Head Pose Angles: Pitch=-2.2226430455762993, Yaw=5.147977808653517, Roll=1.3017287995593352
Sent UDP packet to ('127.0.0.1', 7070): b'\xd8\x13\xe0_\x93\x01\x00\x00\x98\x01\x00\x00\x1c\x01\x00\x00M\x00\x00\x00\xfe\xff\xff\xff'
Total Blinks: 6
Left Eye Center X: 408.5 Y: 284.0
Right Eye Center X: 343.0 Y: 286.0
Left Iris Relative Pos Dx: 79 Dy: -2
Right Iris Relative Pos Dx: -50 Dy: -1

Head Pose Angles: Pitch=-1.92806774114658, Yaw=5.011709475645143, Roll=1.374496573058986
Sent UDP packet to ('127.0.0.1', 7070): b'\x1e\x14\xe0_\x93\x01\x00\x00\x98\x01\x00\x00\x1c\x01\x00\x00O\x00\x00\x00\xfe\xff\xff\xff'
Total Blinks: 6
Left Eye Center X: 408.5 Y: 284.5
Right Eye Center X: 343.0 Y: 286.0
Left Iris Relative Pos Dx: 78 Dy: -2
Right Iris Relative Pos Dx: -50 Dy: -1
```

Наступним кроком протестуємо систему оцінювання знань (рис.3.9).

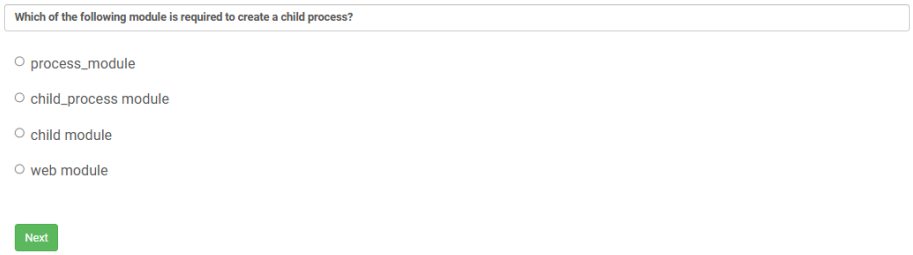


Рисунок 3.9 – Система оцінювання знань

На рисунку 3.9 видно інтерфейс тестування, спробуємо відвести погляд і протестувати реакцію системи (рис.3.10).

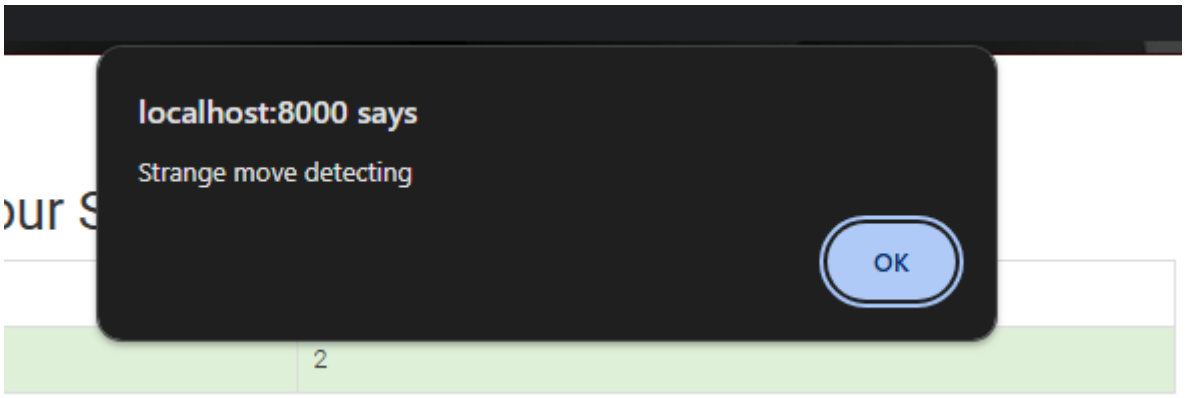


Рисунок 3.10 – Реакція системи на погляд в сторону

Як видно з рисунку 3.10 система відреагувала на погляд в сторону і відправила повідомлення користувачу.

Завершимо тестування і подивимося їх результати (рис.3.11).

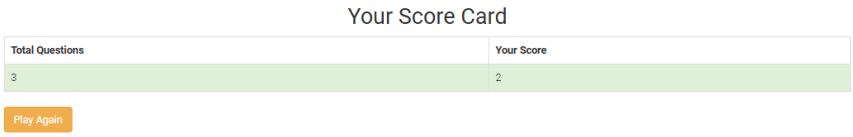


Рисунок 3.11 – Фінальна оцінка знань користувача

Як видно з рисунку 3.11 користувач успішно закінчив тестування і отримав свою оцінку, що означає, що вся система працює коректно і реалізація пройшла успішно.

3.8 Висновки до розділу

У третьому розділі було розроблено і реалізовано програмні модулі, що забезпечують роботу вдосконаленої системи оцінювання знань. На основі сучасних технологій і засобів програмування були створені модулі для керування тестуванням, відстежування рухів і погляду користувача, а також графічний інтерфейс, який дозволяє зручно взаємодіяти із системою. Особливу увагу було приділено інтеграції цих модулів для забезпечення безперервної і надійної роботи системи.

Реалізація модулів відстежування рухів та погляду користувача базувалася на бібліотеках OpenCV та Mediapipe, що дозволило обробляти відеопотоки в реальному часі та виявляти підозрілу поведінку користувача. Це суттєво підвищило рівень контролю під час тестування, запобігаючи можливим спробам шахрайства.

Було також створено модуль для обробки запитань і відповідей, який підтримує динамічне налаштування тестів, зберігає результати і дозволяє адміністраторам легко керувати базою тестів. Завдяки використанню MongoDB та Mongoose було забезпечено швидке зберігання і доступ до даних.

Графічний інтерфейс користувача, реалізований за допомогою Flask, HTML, CSS та JavaScript, забезпечує інтерактивну і зручну взаємодію з системою. Інтерфейс підтримує автоматичне завантаження питань, надсилання відповідей, а також відображення результатів тестування.

Таким чином, у цьому розділі була досягнута головна мета – створення повнофункціональної системи оцінювання знань, яка поєднує високий рівень зручності використання з надійністю контролю. Реалізовані модулі є гнучкими і масштабованими, що дозволяє легко адаптувати систему для різних потреб освітнього процесу.

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Оцінювання комерційного потенціалу розробки програмного забезпечення

Метою проведення комерційного та технологічного аудиту є оцінювання потенціалу вдосконаленої системи оцінювання знань, розробленої на основі модернізованої моделі контролювання дій користувача з використанням технології відслідковування рухів та погляду.

Для виконання технологічного аудиту було залучено трьох незалежних експертів Вінницького національного технічного університету кафедри менеджменту та інформаційної безпеки: доцента, к.т.н., Карпінець В.В., доцента, д.ф., Салієва О.В., професора, д.т.н., Яремчука Ю.Є. Оцінювання проводилося за допомогою таблиці 4.1, у якій за п’ятибальною шкалою на основі 12 критеріїв визначено рівень комерційного потенціалу системи.

Таблиця 4.1 – Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка [41]

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-терій	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів

Продовження таблиці 4.1

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-терій	0	1	2	3	4
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років

Продовження таблиці 4.1

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-терій	0	1	2	3	4
				3-х до 5-ти років	
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки пові-домлення відповідним органам про виробництв о та реалізацію продукту	Відсутні будь-які регламент ні обмеження на виробницт во та реалізацію продукту

Таблиця 4.2 – Рівні комерційного потенціалу розробки

Середньоарифметична сума балів СБ, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0-10	Низький
11-20	Нижче середнього
21-30	Середній
31-40	Вище середнього
41-48	Високий

В таблиці 4.3 наведено результати оцінювання експертами комерційного потенціалу розробки.

Таблиця 4.3 – Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада експерта		
	Яремчук Ю.Є.	Карпінець В.В.	Салієва О.В.
	Бали, виставлені експертами:		
1	5	4	5
2	3	4	3
3	4	5	4
4	4	4	4
5	4	4	4
6	4	3	3
7	4	4	3
8	5	5	4

Продовження таблиці 4.3

Критерії	Прізвище, ініціали, посада експерта		
	Яремчук Ю.Є.	Карпінець В.В.	Салієва О.В.
	Бали, виставлені експертами:		
9	4	4	4
10	3	3	5
11	5	3	4
12	4	3	4
Сума балів	$СБ_1$= 49	$СБ_2$= 46	$СБ_3$= 47
Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_1^3 СБ_i}{3} = \frac{49 + 46 + 47}{3} = 47{,}3$		

Середньоарифметична сума балів, розрахована на основі висновків експертів, склала 47,3 бала, що, згідно з таблицею 4.2, відповідає високому рівню комерційного потенціалу проведених досліджень.

Вдосконалена система оцінювання знань на основі модернізованої моделі контролювання дій користувача з використанням технології відслідковування рухів та погляду – це програмне забезпечення, яке забезпечує контроль за діями користувачів під час тестування. Система дозволяє виявляти несанкціоновані дії, блокувати інтерфейс і підвищувати рівень об’єктивності оцінювання знань.

Ця система може бути корисною для освітніх закладів, компаній, які проводять професійні сертифікації, а також інших організацій, зацікавлених у впровадженні сучасних технологій для контролю знань та зниження ризику списування.

4.2 Прогнозування витрат на виконання наукової роботи та впровадження її результатів

Витрати, пов’язані з виконанням науково-дослідної роботи, розподіляються за такими статтями: оплата праці, витрати на соціальні заходи, закупівля матеріалів, паливо та енергія для науково-виробничих потреб, витрати на службові відрядження, придбання програмного забезпечення для наукової діяльності, інші супутні витрати, а також накладні витрати [41].

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

$$З_О = \frac{М}{Т_р} * t \text{ (грн)}, \quad (4.1)$$

де M – місячний посадовий оклад конкретного розробника (інженера, дослідника, науковця тощо), грн.;

T_P – число робочих днів в місяці; приблизно $T_P \approx 21...23$ дні;

t – число робочих днів роботи дослідника.

$$З_о = \frac{28000 * 32}{22} = 62000 \text{ грн.}$$

Таблиця 4.4 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Менеджер проекту	28 000,0	1 272,7	32	40 727,2
Розробник програмного забезпечення	25 000,0	1 136,4	48	54 545,5
			Всього	95 272,7

Додаткова заробітна плата $З_д$ для всіх розробників та працівників, залучених до створення нового технічного рішення, визначається як 10–12% від їхньої основної заробітної плати. У цьому підприємстві розмір додаткової заробітної плати становить 10% від основної заробітної плати [41].

$$З_д = (З_о + З_р) * \frac{Н_{доп}}{100\%} \quad (4.2)$$

де $Н_{доп}$ – норма нарахування додаткової заробітної плати.

$$З_д = 0{,}1 * 95\ 272{,}7 = 9527{,}3 \text{ (грн)}.$$

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

$$Н_{зп} = (З_о + З_{дод}) * \frac{\beta}{100\%} \quad (4.3)$$

Де β – норма нарахування на заробітну плату.

$$Н_{ЗП} = (95\ 272{,}7 + 9\ 527{,}3\ ) * \frac{22}{100} = 23\ 056\ (грн).$$

Витрати на комплектуючі вироби, які використовують при виготовленні одиниці продукції, розраховуються, згідно їх номенклатури, за формулою:

$$К = \sum_{i=1}^{n} Н_i * Ц_i * К_i\ , \quad (4.4)$$

де $Н_i$ – кількість комплектуючих *i*-го виду, шт.;

$Ц_i$ – покупна ціна комплектуючих *i*-го найменування, грн.;

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

Таблиця 4.5 – Комплектуючі, що використані на розробку

Найменування матеріалу	Ціна за одиницю, грн.	Витрачено	Вартість витраченого матеріалу, грн.
Набір канцелярський	220	2	440
Папір	215	1	215
Папір для заміток	45	1	45
Флешка	140	1	140
		Всього	840
		З врахуванням коефіцієнта транспортування	924

Програмне забезпечення для наукової роботи включає витрати на розробку та придбання спеціальних програмних засобів і програмного забезпечення необхідного для проведення дослідження.

Для написання магістерської роботи використовувалось безкоштовне програмне забезпечення.

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо можуть бути розраховані з використанням прямолінійного методу амортизації за формулою [41]:

$$А_{обл} = \frac{Ц_б}{Т_в} * \frac{t_{вик}}{12} \quad (4.5)$$

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.

$$А_{обл} = \frac{30\,000{,}0 * 2}{3 * 12} = 1\,666{,}6 \text{ грн.}$$

Таблиця 4.6 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Ноутбук Lenovo LOQ 15IAX9I	30 000,0	3	2	1 666,6
Ноутбук Asus Vivobook Go	19 000,0	3	2	1 055,5
Офісне приміщення	12 000,0	5	2	400,0
Оргтехніка	3 500,0	4	2	145,8
			Всього	7 524,6

Витрати на силову електроенергію ($В_е$) розраховують за формулою:

$$В_е = \sum_{i=1}^{п} \frac{W_{уі} * t_і * Ц_е * К_{впі}}{\eta_і} \quad (4.6)$$

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

t_i – тривалість роботи обладнання на етапі дослідження, год;

$Ц_е$ – вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії), приймемо $Ц_е$ = 7,50 грн;

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

η_i – коефіцієнт корисної дії обладнання, $\eta_i < 1$

$$В_е = \frac{0{,}3 * 256 * 7{,}50 * 0{,}95}{0{,}97} = 564{,}1 \text{ грн.}$$

Таблиця 4.7 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Ноутбук Lenovo LOQ 15IAX9I	0,3	256	564,1
Ноутбук Asus Vivobook Go	0,3	384	846,2
Офісне приміщення	0,3	384	846,2
Оргтехніка	0,2	20	29,4
Всього			2 285,9

Витрати за статтею «Службові відрядження» розраховуються як 20…25% від суми основної заробітної плати дослідників та робітників за формулою:

$$В_{св} = (З_о + З_р) * \frac{Н_{св}}{100\%} \quad (4.7)$$

де $Н_{св}$ – норма нарахування за статтею «Службові відрядження».

$$В_{св} = (95\ 272{,}7) * \frac{25}{100} = 23\ 818{,}2 \text{ грн.}$$

Витрати за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації» розраховуються як 30…45% від суми основної заробітної плати дослідників та робітників за формулою:

$$В_{сп} = (З_о + З_р) * \frac{Н_{сп}}{100\%} \quad (4.8)$$

де $Н_{сп}$ – норма нарахування за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації».

$$В_{сп} = (95\ 272{,}7) * \frac{35}{100} = 33\ 345{,}4 \text{ грн.}$$

Витрати за статтею «Інші витрати» розраховуються як 50…100% від суми основної заробітної плати дослідників та робітників за формулою:

$$І_{в} = (З_{о} + З_{р}) * \frac{Н_{ів}}{100\%} \quad (4.9)$$

де $Н_{ів}$ – норма нарахування за статтею «Інші витрати».

$$І_{в} = (95\ 272{,}7) * \frac{55}{100} = 52\ 399{,}9 \text{ грн.}$$

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

$$В_{нзв} = (З_{о} + З_{р}) * \frac{Н_{нзв}}{100\%} \quad (4.10)$$

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

$$В_{нзв} = (95\ 272{,}7) * \frac{100}{100} = 95\ 272{,}7 \text{ грн.}$$

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

$$В_{заг} = З_{о} + З_{р} + З_{дод} + З_{н} + М + В_{прг} + А_{обл} + В_{е} + В_{св} + В_{сп} + І_{в} + В_{нзв} \quad (4.11)$$

$$В_{заг} = 95\ 272{,}7 + 9\ 527{,}3 + 23\ 056 + 924 + 0 + 7\ 524{,}6 + 2\ 285{,}9 + 23\ 818{,}2 + 33\ 345{,}4 + 52\ 399{,}9 + 95\ 272{,}7 = 343\ 426{,}7 \text{ грн.}$$

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

$$ЗВ = \frac{В_{заг}}{\eta} \quad (4.12)$$

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 [42].

$$ЗВ = \frac{343\ 426,7}{0,7} = 490\ 609,5 \text{ грн.}$$

4.3 Прогнозування комерційних ефектів від реалізації результатів розробки

В ринкових умовах основним позитивним результатом для потенційного інвестора від впровадження результатів науково-технічної розробки є зростання чистого прибутку.

Дослідження за темою «Вдосконалена система оцінювання знань на основі моделі контролювання дій користувача з використанням технології відслідковування рухів та погляду» передбачають її комерціалізацію протягом трьох років на ринку. Майбутній економічний ефект у цьому випадку буде формуватися на основі таких даних:

Збільшення кількості споживачів продукту в аналізовані періоди часу завдяки покращенню його характеристик (ΔN):

1-й рік – 65 користувачів;

2-й рік – 55 користувачів;

3-й рік – 45 користувачів.

Кількість споживачів, які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки, становить 100 користувачів (N).

Вартість програмного продукту до впровадження результатів розробки – 100 000 грн ($Ц_б$).

Зміна вартості програмного продукту від впровадження результатів науково-технічної розробки – 1 000 грн ($\pm ΔЦ_б$).

Можливе збільшення чистого прибутку потенційного інвестора для кожного з трьох років, протягом яких очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, розраховується за формулою.

$$\Delta П_i = (\pm \Delta Ц_б * N + Ц_б * \Delta N)_i * \lambda * \rho * (1 - \frac{\vartheta}{100}) \quad (4.13)$$

де λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2024 році ставка податку на додану вартість складає 20%, а коефіцієнт λ *=0,8333*;

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту. Приймемо ρ =25*%*;

ϑ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2024 році ϑ *=18%*;

1-й рік: $\Delta П_1 = (1000 \times 100 + 100\,000 \times 65) \times 0{,}83 \times 0{,}25 \times \left(1 - \frac{0{,}18}{100}\right) = 13\,483\,935{,}2$ (грн.)

2-й рік: $\Delta П_2 = (1000 \times 100 + 100\,000 \times (65 + 55)) \times 0{,}83 \times 0{,}25 \times \left(1 - \frac{0{,}18}{100}\right) = 25\,875\,892{,}7$ (грн.)

3-й рік: $\Delta П_3 = (1000 \times 100 + 100\,000 \times (65 + 55 + 45)) \times 0{,}83 \times 0{,}25 \times \left(1 - \frac{0{,}18}{100}\right) = 34\,196\,585{,}2$ (грн.)

Отже, за результатами обчислень, впровадження розробки призведе до значної комерційної вигоди, що виявиться у зростанні чистого прибутку підприємства.

4.4 Розрахунок ефективності вкладених інвестицій та періоду їх окупності

Ключовими критеріями, що впливають на рішення інвестора щодо фінансування наукової розробки, є абсолютна та відносна ефективність інвестицій, а також термін їх окупності.

На початковому етапі визначається теперішня вартість інвестицій (PV), які будуть спрямовані на наукову розробку.

Також розраховується обсяг початкових вкладень, які потенційний інвестор повинен здійснити для впровадження та комерціалізації науково-технічного проєкту [42].

$$\text{PV} = k_{інв} * ЗВ \tag{4.14}$$

$k_{інв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо $k_{інв}$=2;

$ЗВ$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 490 609,5 грн.

$$\text{PV} = 2 * 490\,609{,}5 = 981\,219 \text{ грн.}$$

Розрахуємо абсолютну ефективність вкладених інвестицій $E_{абс}$ згідно наступної формули:

$$\text{Е}_{\text{абс}} = (\text{ПП} - \text{PV}) \tag{4.15}$$

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

PV – теперішня вартість початкових інвестицій, грн.

Приведена вартість всіх чистих прибутків ПП розраховується за формулою:

$$\text{ПП} = \sum_1^{\text{Т}} \frac{\Delta\text{П}_1}{(1+\tau)^t} \tag{4.16}$$

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

Т – 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.

$$ПП = \frac{13\,483\,935{,}2}{(1+0{,}2)^1} + \frac{25\,875\,892{,}7}{(1+0{,}2)^2} + \frac{34\,196\,585{,}2}{(1+0{,}2)^3} = 48\,995\,673{,}1 \text{ грн.}$$

$$E_{абс} = 48\,995\,673{,}1 - 981\,219 = 48\,014\,454{,}1 \text{ грн.}$$

Оскільки $E_{абс} > 0$, встановлено, що проведення наукових досліджень для розробки програмного продукту та його подальше впровадження принесуть прибуток. Це підтверджує доцільність проведення досліджень [42].

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

$$E_в = \sqrt[Т_ж]{1 + \frac{E_{абс}}{PV}} - 1 \qquad (4.17)$$

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

PV – теперішня вартість початкових інвестицій, грн;

$Т_ж$ – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримування позитивних результатів від її впровадження, роки.

$$E_в = \sqrt[з]{1 + \frac{48\,014\,454{,}1}{981\,219}} - 1 = 2{,}6$$

Порівняємо Ев з мінімальною (бар'єрною) ставкою дисконтування τ_{min}, яка визначає мінімальну дохідність, нижче якої інвестиції не будуть здійснюватися.

У загальному вигляді мінімальна (бар'єрна) ставка дисконтування τ_{min} визначається за формулою:

$$\tau_{min} = d + f \qquad (4.18)$$

d – середньозважена ставка за депозитними операціями в комерційних банка;

f – показник, що характеризує ризикованість вкладень; *f* = 0,4.

d = 0,2.

$$\tau_{min} = 0{,}2 + 0{,}4 = 0{,}6$$

Оскільки $E_в$ = 260% > τ_{min} = 60%, то у інвестора є потенційна зацікавленість у фінансуванні даної наукової розробки.

Далі розраховуємо період окупності інвестицій $Т_{ок}$, які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки [42]:

$$Т_{ок} = \frac{1}{Е_в} \quad (4.19)$$

де $Е_в$ – внутрішня економічна дохідність вкладених інвестицій.

$$Т_{ок} = \frac{1}{2{,}6} = 0{,}3$$

Якщо $Т_{ок}$ < 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.

4.5 Висновки до розділу

Проведено оцінку комерційного потенціалу вдосконаленої системи оцінювання знань на основі моделі контролювання дій користувача з використанням технології відслідковування рухів та погляду. Ця система дозволяє проводити контроль за діями користувачів, автоматизувати процес виявлення несанкціонованих дій та покращити об’єктивність оцінювання знань.

Прогнозовані витрати на виконання науково-дослідної роботи за окремими статтями становлять 343 426,7 грн, а загальна сума витрат на розробку та впровадження системи - 490 609,5 грн.

Інвестиції, вкладені в цей проєкт, окупляться через 0,3 роки. Приведена вартість сукупного чистого прибутку, який отримає підприємство від комерціалізації результатів цієї розробки, складає 48 995 673,1 грн.

ВИСНОВКИ

У ході виконання магістерської роботи була розроблена вдосконалена система оцінювання знань, яка забезпечує високий рівень контролю та чесності під час тестування за рахунок інтеграції технологій відстежування рухів і погляду користувача. Основною метою роботи було створення системи, яка б поєднувала сучасні технології моніторингу поведінки користувача з адаптивними моделями оцінювання знань, що дозволяє мінімізувати ризики шахрайства та підвищити ефективність освітніх процесів.

Під час виконання роботи були досягнуті наступні результати:

Проведено огляд сучасних підходів до оцінювання знань та методів контролювання дій користувача, що дозволило визначити ключові проблеми існуючих систем і сформулювати основні вимоги до вдосконаленої системи.

- розроблено алгоритми інтеграції технологій відстежування рухів та погляду користувача, які дозволяють у реальному часі аналізувати поведінкові патерни користувача і виявляти підозрілі дії.

- реалізовано програмний модуль моніторингу, що базується на бібліотеках OpenCV та Mediapipe, для аналізу рухів користувача, а також модуль відстежування погляду, що дозволяє фіксувати напрямок уваги користувача під час тестування.

- розроблено модуль для керування тестуванням, який підтримує динамічне налаштування тестів, зберігання результатів у базі даних MongoDB та автоматичний аналіз відповідей користувачів.

- створено графічний інтерфейс користувача для інтерактивного взаємодії із системою, що забезпечує зручність та простоту використання.

Запропонована система є інноваційною у сфері автоматизованого оцінювання знань, оскільки впроваджує методи моніторингу поведінки, що раніше не були поширеними у подібних системах. Вона забезпечує підвищення точності та об’єктивності оцінювання, зменшення впливу шахрайства та можливість аналізу як академічних результатів, так і поведінкових характеристик користувачів.

Результати роботи підтверджують, що інтеграція технологій моніторингу поведінки з адаптивними моделями оцінювання знань є ефективним підходом до вдосконалення сучасних освітніх систем. Подальший розвиток роботи може включати масштабування системи для використання у великих навчальних установах, інтеграцію з платформами дистанційного навчання та розширення функціоналу за рахунок машинного навчання для точнішого аналізу поведінкових даних.

Таким чином, розроблена система є важливим внеском у підвищення якості та чесності процесу оцінювання знань, що сприятиме розвитку інноваційних технологій у сфері освіти.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Загрози і вразливості. StudFiles. URL: https://studfile.net/preview/9094212/page:4/ (дата звернення: 18.10.2024).

2. Засоби та методи захисту інформації - бібліотека buklib.net. Головна - Бібліотека BukLib.net. URL: https://buklib.net/books/28625/ (дата звернення: 16.10.2024).

3. Захист інформації від несанкціонованого доступу, Методи та види НСД з курсу Захист та безпека інформаційних ресурсів, НУДПСУ. URL: https://ua.kursoviks.com.ua/metodychni_vkazivky/article_post/791-lektsiya-11-na-temu-zakhist-informatsii-vid-nesanktsionovanogo-dostupu-metodi-ta-vidi-nsd-z-kursu-zakhist-ta-bezpeka-informatsiynikh-resursiv-nudpsu (дата звернення: 16.10.2024).

4. Захист інформації від несанкціонованого доступу. Stud. URL: https://stud.com.ua/43315/informatika/zahist_informatsiyi_nesanktsionovanogo_dostupu (дата звернення: 16.10.2024).

5. Захист інформації в інформаційних системах. Pidru4niki. URL: https://pidru4niki.com/13670622/informatika/zahist_informatsiyi_informatsiynih_sistemah (дата звернення: 16.10.2024).

6. Захист інформації. Енциклопедія Сучасної України. URL: https://esu.com.ua/article-15872 (дата звернення: 17.10.2024).

7. Detecting and responding to unauthorized access. Code42. URL: https://www.code42.com/blog/detecting-and-responding-to-unauthorized-access/ (date of access: 16.10.2024).

8. How to protect your data from unauthorized access. CYPRESS DATA DEFENSE | Cypress Data Defense. URL: https://www.cypressdatadefense.com/blog/unauthorized-data-access/ (date of access: 17.10.2024).

9. Yagiz Kaymak. Tracking user application activity by using machine learning techniques on network traffic. Конференція: 2019 Міжнародна конференція зі штучного інтелекту в інформації та комунікації (ICAIIC), Okinawa, 19 February 2019. 2019.

10. Класифікація методів виявлення аномалій в інформаційних системах. ElAr :: Головна. URL: https://openarchive.nure.ua/bitstreams/7c434471-942c-40a7-b70c-0cc2655a42fe/download (дата звернення: 18.10.2024).

11. Гладка В. АНАЛІЗ ІСНУЮЧИХ СИСТЕМ КОНТРОЛЮВАННЯ ДІЙ КОРИСТУВАЧА [Електронний ресурс] / В. Гладка, О. В. Салієва // ВНТУ. – 2024. – Режим доступу до ресурсу: https://conferences.vntu.edu.ua/index.php/mn/mn2025/author/submission/22816.

12. Методи блокувань. URL: https://web.posibnyky.vntu.edu.ua/fitki/11petuh_bazdanyh_movy_zalitiv/25.htm (дата звернення: 18.10.2024).

13. Что такое SIEM система?. URL: https://ua.softlist.com.ua/articles/chto-takoe-siem-sistema/ (дата звернення: 18.10.2024).

14. Що таке "Інформація про безпеку та управління подіями" (SIEM)? | Gridinsoft. ТОВ "Грідінсофт". URL: https://gridinsoft.ua/siem (дата звернення: 18.10.2024).

15. Що таке керування вразливостями?. microsoft. URL: https://www.microsoft.com/uk-ua/security/business/security-101/what-is-vulnerability-management (дата звернення: 18.10.2024).

16. All-in-one security, identity, and privacy protection. McAfee. URL: https://www.mcafee.com/ (date of access: 17.10.2024).

17. Why developers like McAfee Endpoint Protection. StackShare. URL: https://stackshare.io/mcafee-endpoint-protection (date of access: 17.10.2024).

18. Daisy. What is microsoft defender advanced threat protection and how to use it. EaseUS. URL: https://www.easeus.com/knowledge-center/what-is-microsoft-defender-advanced-threat-protection.html (date of access: 17.10.2024).

19. Johnson J. T. Review Microsoft Defender for endpoint security pros and cons | TechTarget. Security. URL: https://www.techtarget.com/searchsecurity/tip/Review-Microsoft-Defender-for-endpoint-security-pros-and-cons (date of access: 17.10.2024).

20. Технології контролювання дій користувача у системах оцінювання знань. URL: https://itpro.ua/tech/user-activity-monitoring (дата звернення: 24.11.2024).

21. Що таке системи моніторингу користувачів? Огляд та можливості. URL: https://monitoring-tools.ua/overview (дата звернення: 24.11.2024).

22. Використання Eye-Tracking у сучасних освітніх системах. URL: https://education-innovations.com/eye-tracking (дата звернення: 24.11.2024).

23. Motion-tracking технології в освіті. URL: https://motionedu.org/tech/motion-tracking-in-education (дата звернення: 24.11.2024).

24. Інтеграція контролю дій користувача з системами тестування. URL: https://integrationplatforms.com/user-monitoring (дата звернення: 24.11.2024).

25. Використання технологій відстежування у захисті від шахрайства. URL: https://fraudtech.com.ua/tracking-technologies (дата звернення: 24.11.2024).

26. Алгоритми контролювання дій користувача у реальному часі. URL: https://algorithmhub.ua/user-action-control (дата звернення: 24.11.2024).

27. Застосування OpenCV у системах моніторингу. URL: https://opencv.org.ua/applications (дата звернення: 24.11.2024).

28. Mediapipe: Інструмент для відстежування рухів. URL: https://mediapipe.dev/solutions (дата звернення: 24.11.2024).

29. Сучасні методи адаптивного тестування. URL: https://adaptive-tests.org/methods (дата звернення: 24.11.2024).

30. Моделі прийняття рішень у системах оцінювання знань. URL: https://decisionmodels.edu.com (дата звернення: 24.11.2024).

31. Порівняння технологій Eye-Tracking та Motion-Tracking. URL: https://techcompare.ua/eye-vs-motion (дата звернення: 24.11.2024).

32. Використання MongoDB у системах моніторингу. URL: https://mongodb.com/use-cases (дата звернення: 24.11.2024).

33. Інтеграція Flask у освітніх системах. URL: https://flaskedu.org/integration (дата звернення: 24.11.2024).

34. Python як основа для систем тестування. URL: https://python.org/applications/testing (дата звернення: 24.11.2024).

35. Протидія шахрайству за допомогою поведінкового аналізу. URL: https://fraudanalytics.com.ua/behavior-analysis (дата звернення: 24.11.2024).

36. Переваги використання OpenCV для моніторингу користувачів. URL: https://opencvapplications.com/user-monitoring (дата звернення: 24.11.2024).

37. Графічний інтерфейс для систем оцінювання знань. URL: https://uiuxedu.org/testing-interface (дата звернення: 24.11.2024).

38. Тестування програмного забезпечення для навчальних платформ. URL: https://softwaretesting.edu.com (дата звернення: 24.11.2024).

39. Перспективи впровадження AI в освітні системи. URL: https://aieducation.org/ai-systems (дата звернення: 24.11.2024).

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

41. Кавецький В. В. Економічне обґрунтування інноваційних рішень: практикум / В. В. Кавецький, В. О. Козловський, І. В. Причепа. Вінниця : ВНТУ, 2016. 113 с

ДОДАТКИ

116

Додаток А. Технічне завдання

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 2024 р.

ТЕХНІЧНЕ ЗАВДАННЯ

до магістерської кваліфікаційної роботи на тему:

Вдосконалена система оцінювання знань на основі вдосконаленої моделі контролювання дій користувача з використанням технології відслідковування рухів та погляду користувача

08-72.МКР.004.00.134.ТЗ

Керівник магістерської кваліфікаційної роботи

д. ф., доцент Салієва О.В.

Вінниця – 2024 р.

1. Найменування та область застосування

Програмний засіб вдосконаленя системи оцінювання знань на основі вдосконаленої моделі контролювання дій користувача з використанням технології відслідковування рухів та погляду користувача

2.Підстава для розробки

Розробка виконується на основі наказу ректора ВНТУ №310 від 17. 09. 2024 р.

3. Мета та призначення розробки

3.1 Мета розробки: розробка ефективного методу захисту від атак

3.2 Призначення: розроблений програмний засіб виконує захист від атак

4. Джерела розробки

4.1. Abdelfattah, A., Shrestha, A., Coffey, M., & Cerny, T. (2022). "Systematic Review of Authentication and Authorization Advancements for the Internet of Things." *Sensors*, *22*(4), 1361.

4.2. IEEE Xplore. (2024). "Multi-Factor Authentication in Cyber Physical Systems: A State of Art Survey." *IEEE Conference Publication*.

4.3. IEEE Xplore. (2019). "A Survey on Biometric Authentication: Toward Secure and Privacy-Preserving Identification." *IEEE Journals & Magazine*.

4.4. MDPI Sensors. (2021). "Advancements in Access Control Models and Security Mechanisms for Data Systems." *Sensors*, *21*(7).

5. Вимоги до програми

5.1 Вимоги до функціональних характеристик:

5.1.1 Програмний засіб повинен мати зручний, легкий у використанні інтерфейс користувача;

5.1.2 Реалізація методу не повинна вимагати спеціальних ліцензійних програмних додатків;

5.1.3 Програмний засіб повинен виконувати процес автентифікації користувачів у системі.

5.2 Вимоги до надійності:

5.2.1 Програмний засіб повинен працювати без помилок, у випадку виникнення критичних ситуацій необхідно передбачити виведення відповідних повідомлень;

5.2.2 Бази даних повинні бути налаштовані на автоматичне створення резервних копій;

5.2.3 Програмний засіб повинен виконувати свої функції.

5.3 Вимоги до складу і параметрів технічних засобів:

– процесор – Pentium 1500 МГц і подібні до них;

– оперативна пам’ять – не менше 512 Mb;

– середовище функціонування – операційна система сімейство Windows;

– вимоги до техніки безпеки при роботі з програмою повинні відповідати існуючим вимогам та стандартам з техніки безпеки при користуванні комп’ютерною технікою.

6. Вимоги до програмної документації

6.1 Обов’язкова поетапна інструкція для майбутніх користувачів, наведена у пункті 3.4

7. Вимоги до технічного захисту інформації

7.1 Необхідно забезпечити захист розроблюваного програмного засобу від несанкціонованого використання.

7.2 Неможливість отримання доступу незареєстрованих користувачів до інформаційних ресурсів.

8. Техніко-економічні показники

8.1 Цінність результатів використання даного проекту повинна перевищувати витрати на його реалізацію.

8.2 Має бути реалізований таким чином, щоб підходити для використання широкого загалу.

9. Стадії та етапи розробки

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Початок	Закінчення
1.	Визначення напрямку магістерської роботи, формулювання теми	20.09.2024	31.09.2024
2.	Аналіз предметної області обраної теми	01.10.2024	15.10.2024
3.	Розробка роботи	16.10.2024	26.10.2024
4.	Написання магістерської роботи на основі розробленої теми	27.10.2024	15.11.2024
5.	Передзахист магістерської кваліфікаційної роботи	16.11.2024	24.11.2024
6.	Виправлення, уточнення, корегування магістерської кваліфікаційної роботи	27.11.2024	04.12.2024
7.	Захист магістерської кваліфікаційної роботи	10.12.2024	10.12.2024

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  Гладка В.

Додаток Б. Лістинг програми

```
import cv2
import mediapipe as mp
import requests

# Ініціалізація Mediapipe Pose
mp_pose = mp.solutions.pose
pose = mp_pose.Pose()
mp_drawing = mp.solutions.drawing_utils

# Захоплення відеопотоку
cap = cv2.VideoCapture(0)

if not cap.isOpened():
    print("Error: Cannot access the camera")
    exit()

while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
        break

    # Перетворення кадру для обробки
    frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    results = pose.process(frame_rgb)

    if results.pose_landmarks:
        # Візуалізація ключових точок
        mp_drawing.draw_landmarks(frame, results.pose_landmarks,
mp_pose.POSE_CONNECTIONS)

        # Аналіз координат голови
        nose = results.pose_landmarks.landmark[mp_pose.PoseLandmark.NOSE]
        print(f"Голова: X={nose.x}, Y={nose.y}, Z={nose.z}")

        # Виявлення підозрілої поведінки
        if nose.x < 0.3 or nose.x > 0.7:
            print("Підозріла поведінка: сильний нахил голови")
            data = {
                "gaze_direction": [nose.x, nose.y, nose.z],
                "head_movement": "tilt"
            }
            response = requests.post("http://localhost:5000/track-user-behavior", json=data)
            print(response.json())

    # Відображення кадру
    cv2.imshow('Motion Tracking', frame)

    # Вихід за натисканням "q"
    if cv2.waitKey(10) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
import cv2
import requests
```

```
        # Завантаження моделей для розпізнавання обличчя та очей
        face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 
'haarcascade_frontalface_default.xml')
        eye_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_eye.xml')

        # Захоплення відеопотоку
        cap = cv2.VideoCapture(0)

        if not cap.isOpened():
            print("Error: Cannot access the camera")
            exit()

        while cap.isOpened():
            ret, frame = cap.read()
            if not ret:
                break

            # Перетворення кадру в чорно-білий формат
            gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

            # Виявлення обличчя
            faces = face_cascade.detectMultiScale(gray, scaleFactor=1.3, minNeighbors=5)
            for (x, y, w, h) in faces:
                cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 0, 0), 2)

                # Виявлення очей
                roi_gray = gray[y:y + h, x:x + w]
                roi_color = frame[y:y + h, x:x + w]
                eyes = eye_cascade.detectMultiScale(roi_gray)
                for (ex, ey, ew, eh) in eyes:
                    cv2.rectangle(roi_color, (ex, ey), (ex + ew, ey + eh), (0, 255, 0), 2)

                    # Аналіз напрямку погляду
                    eye_center_x = ex + ew // 2
                    face_center_x = w // 2

                    if eye_center_x < face_center_x - 20:
                        direction = "left"
                        print("Погляд ліворуч")
                    elif eye_center_x > face_center_x + 20:
                        direction = "right"
                        print("Погляд праворуч")
                    else:
                        direction = "center"
                        print("Погляд прямо")

                    # Надсилання даних на сервер
                    data = {"gaze_direction": direction}
                    response = requests.post("http://localhost:5000/track-user-behavior", json=data)
                    print(response.json())

            # Відображення кадру
            cv2.imshow('Eye Tracking', frame)

            # Вихід за натисканням "q"
            if cv2.waitKey(10) & 0xFF == ord('q'):
                break
```

```
cap.release()
cv2.destroyAllWindows()
import tkinter as tk
from tkinter import messagebox

# Питання для тесту
questions = [
   {"text": "What is the capital of France?", "options": ["Paris", "London", "Berlin"], "correct":
0},
   {"text": "What is 2 + 2?", "options": ["3", "4", "5"], "correct": 1}
]

class QuizApp:
   def __init__(self, root):
      self.root = root
      self.root.title("Knowledge Test")
      self.current_question = 0
      self.score = 0

      self.question_label = tk.Label(root, text="", font=("Arial", 16), wraplength=400)
      self.question_label.pack(pady=20)

      self.options = []
      for i in range(3):
         btn = tk.Radiobutton(root, text="", value=i, variable=tk.IntVar())
         btn.pack(anchor="w")
         self.options.append(btn)

      self.next_button = tk.Button(root, text="Next", command=self.next_question)
      self.next_button.pack(pady=20)

      self.show_question()

   def show_question(self):
      question = questions[self.current_question]
      self.question_label.config(text=question["text"])
      for i, option in enumerate(question["options"]):
         self.options[i].config(text=option)

   def next_question(self):
      selected_option = None
      for i, btn in enumerate(self.options):
         if btn.var.get() == 1:
            selected_option = i

      if selected_option == questions[self.current_question]["correct"]:
         self.score += 1

      self.current_question += 1
      if self.current_question < len(questions):
         self.show_question()
      else:
         messagebox.showinfo("Test Completed", f"Your score: {self.score}/{len(questions)}")
         self.root.quit()

root = tk.Tk()
app = QuizApp(root)
root.mainloop()@app.route('/get-questions', methods=['GET'])
```

```
def get_questions():
    # Приклад питань для тесту
    questions = [
        {
            "id": 1,
            "text": "What is the capital of France?",
            "options": [
                {"id": 1, "text": "Paris"},
                {"id": 2, "text": "London"},
                {"id": 3, "text": "Berlin"}
            ]
        },
        {
            "id": 2,
            "text": "What is 2 + 2?",
            "options": [
                {"id": 4, "text": "3"},
                {"id": 5, "text": "4"},
                {"id": 6, "text": "5"}
            ]
        }
    ]
    return jsonify(questions)

@app.route('/start-test', methods=['POST'])
def start_test():
    data = request.json  # Отримуємо відповіді користувача
    correct_answers = {1: 1, 2: 5}  # Правильні відповіді
    score = 0

    for question_id, selected_option in data['answers'].items():
        if correct_answers.get(int(question_id)) == selected_option:
            score += 1

    return jsonify({"message": "Test completed", "score": score})<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Knowledge Test</title>
    <style>
        body { font-family: Arial, sans-serif; margin: 20px; }
        h1 { text-align: center; }
        .question { margin-bottom: 20px; }
        .option { margin-left: 20px; }
    </style>
</head>
<body>
    <h1>Knowledge Test</h1>
    <div id="questions-container"></div>
    <button id="submit-btn" style="display: none;">Submit</button>

    <script>
        // Завантаження питань із сервера
        fetch('/get-questions')
            .then(response => response.json())
            .then(data => {
                const container = document.getElementById('questions-container');
```

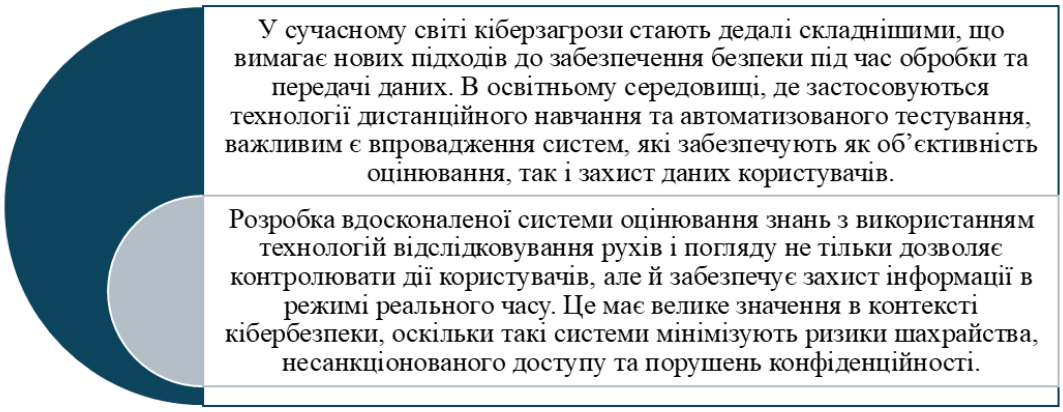
```
            data.forEach((question, index) => {
              const questionDiv = document.createElement('div');
              questionDiv.classList.add('question');
              questionDiv.innerHTML = `<p><strong>${index + 1}. ${question.text}</strong></p>`;
              question.options.forEach(option => {
                questionDiv.innerHTML += `
                  <div class="option">
                    <input type="radio" name="question-${question.id}" value="${option.id}">
                    <label>${option.text}</label>
                  </div>`;
              });
              container.appendChild(questionDiv);
            });

            // Відображення кнопки "Submit", якщо є питання
            if (data.length > 0) {
              document.getElementById('submit-btn').style.display = 'block';
            }
          });

        // Надсилання відповідей користувача
        document.getElementById('submit-btn').addEventListener('click', () => {
          const answers = {};
          const inputs = document.querySelectorAll('input[type="radio"]:checked');
          inputs.forEach(input => {
            const questionId = input.name.split('-')[1];
            answers[questionId] = parseInt(input.value);
          });

          fetch('/start-test', {
            method: 'POST',
            headers: { 'Content-Type': 'application/json' },
            body: JSON.stringify({ answers })
          })
          .then(response => response.json())
          .then(data => alert(`Your score: ${data.score}`));
        });
      </script>
    </body>
    </html>
```

Додаток В. Ілюстративний матеріал



Актуальність та мета

Актуальність

Актуальність теми полягає в необхідності забезпечення чесності та об'єктивності під час оцінювання знань, що є важливим у сучасному освітньому процесі. Використання технологій відслідковування рухів та погляду дозволяє мінімізувати ризики несанкціонованої поведінки, таких як списування, та підвищити надійність результатів тестування. Це відповідає сучасним тенденціям у цифровізації освіти та впровадженню інноваційних технологій у навчальний процес.

Мета

Метою дослідження є розробка вдосконаленої системи оцінювання знань, яка інтегрує технології моніторингу рухів і погляду користувача для підвищення точності оцінювання та зниження ризиків шахрайства під час тестування.

Відслідковування рухів (Motion Tracking)

Відслідковування рухів (Motion Tracking) — це технологія, яка дозволяє фіксувати й аналізувати переміщення об'єктів або користувачів у просторі за допомогою спеціальних датчиків, камер або сенсорів. У сфері освітніх технологій ця система застосовується для моніторингу поведінки користувачів під час тестування, щоб запобігти несанкціонованим діям, таким як списування або використання сторонніх матеріалів.

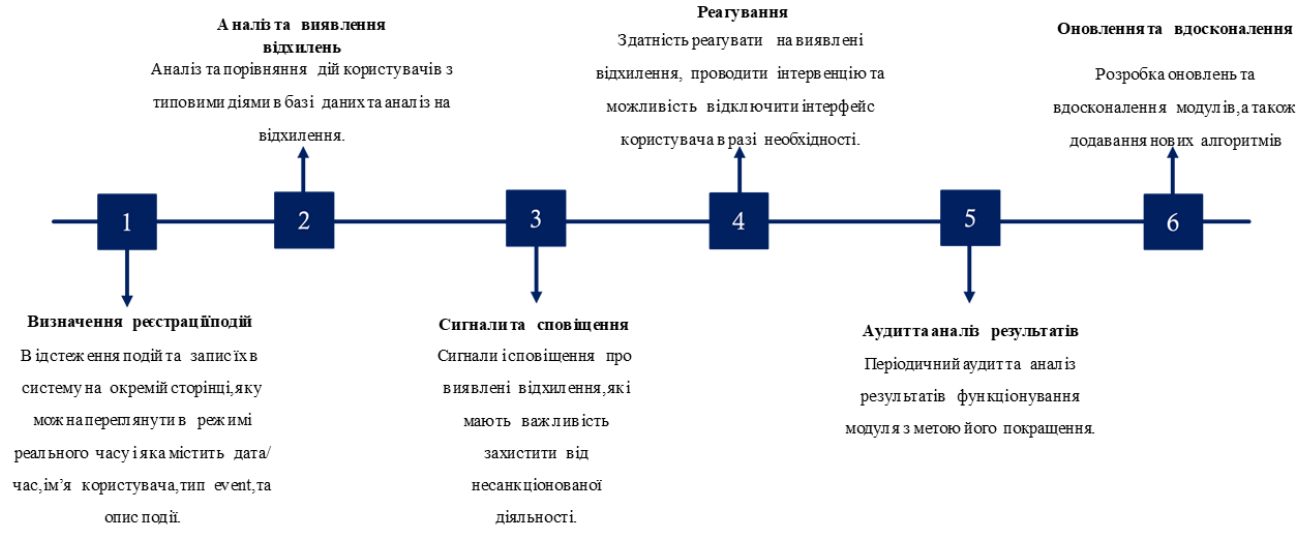
Схема роботи системи motion-tracking з основними етапами

Відслідковування погляду (Gaze Tracking)

Відслідковування погляду (Gaze Tracking) — це технологія, яка дозволяє визначати, куди користувач дивиться в певний момент часу. Вона базується на відстеженні рухів очей, щоб аналізувати, на яких об'єктах або ділянках екрану зосереджена увага користувача. Ця технологія широко використовується в різних галузях, включаючи кібербезпеку, медицину, дослідження поведінки та оцінювання знань.

Схема роботи технології відслідковування погляду

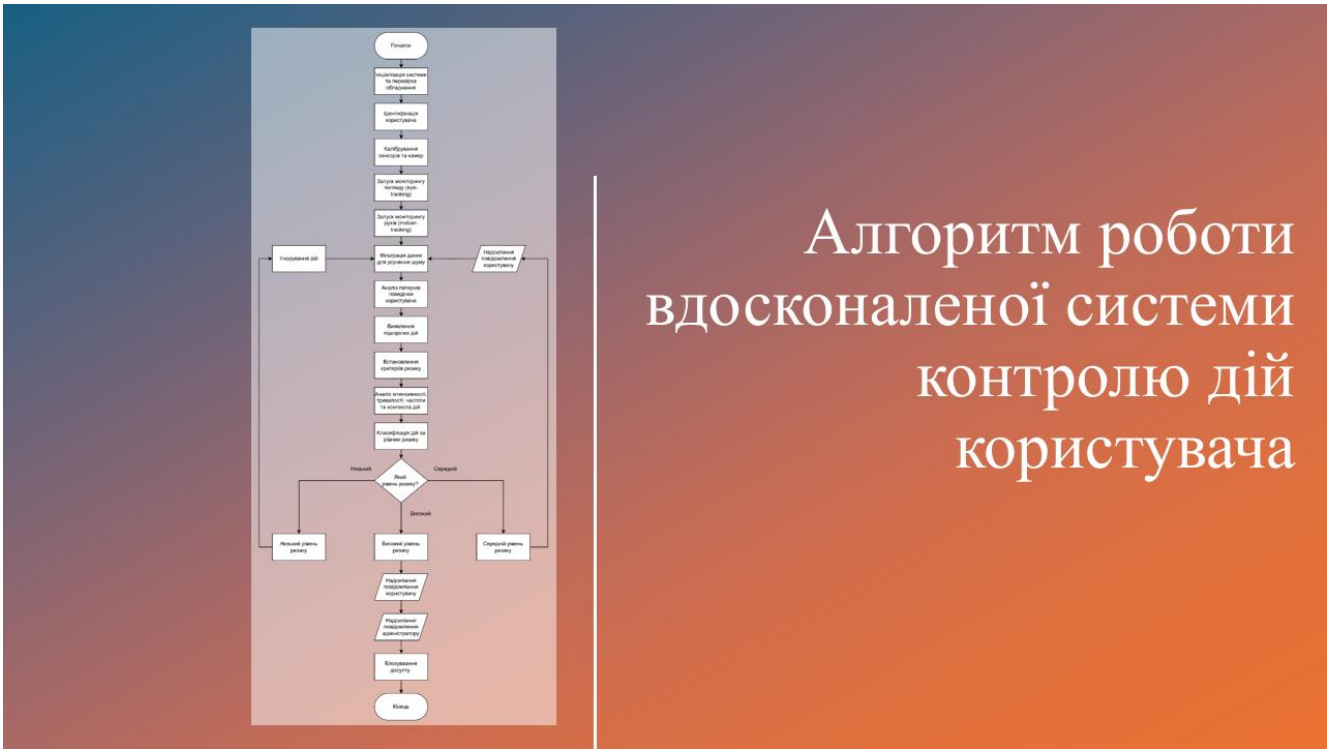
Алгоритми вдслідковування несанкціонаваних дій



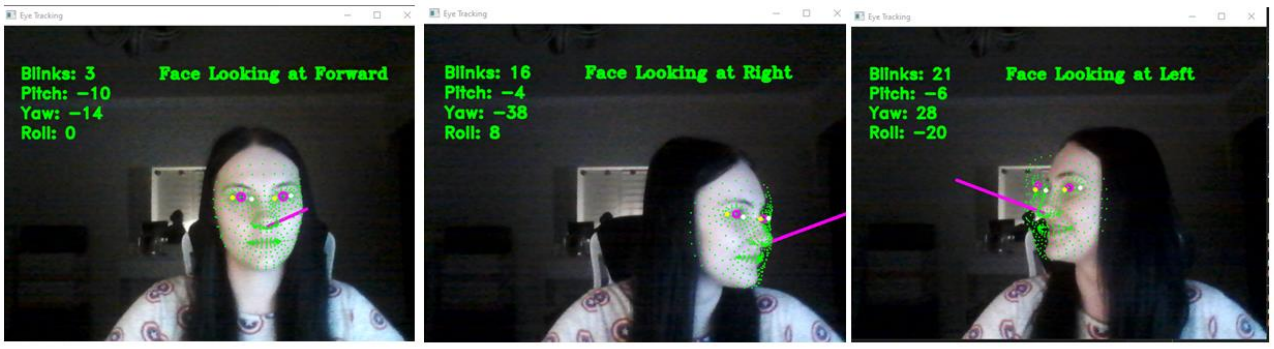
Порівняння з аналогами

Характеристика	Proctorio	Respondus	Examity	ProctorU
Тип проктерингу	Автоматизований або живий, з можливістю вибору рівня контролю.	Автоматизований (LockDown Browser) та живий (Monitor).	Живий, автоматизований або гібридний підхід.	Живий проктеринг із автоматизованим моніторингом.
Моніторинг оточення	Аналіз відеопотоку для виявлення сторонніх осіб або заборонених матеріалів.	Забезпечується через LockDown Browser та відеомоніторинг (Monitor).	Використовує вебкамеру для запису відеопотоку та моніторингу оточення.	Виконується через постійний відеозапис і спостереження.
Автентифікація	Використовує багатофакторну перевірку: фото, ID, біометрія.	Перевірка особи через вебкамеру перед початком тесту.	Включає ідентифікацію за фото, перевірку документів та біометрію.	Фото, документи та перевірка особи проктором у реальному часі.
Інтеграція з LMS	Повна інтеграція з більшістю популярних LMS (Canvas, Moodle, Blackboard тощо).	Інтегрується з усіма популярними LMS.	Легка інтеграція з LMS, можливість масштабування для різних платформ.	Інтегрується з популярними LMS, включаючи Canvas, Moodle та Blackboard.
Конфіденційність	Наскрізне шифрування даних для забезпечення безпеки. Низькі вимоги до пропускної здатності інтернету.	Дані зберігаються локально або передаються зашифрованими.	Високий рівень захисту даних, однак можуть виникати питання щодо збереження конфіденційності користувачів.	Застосовує політики конфіденційності, але потребує зберігання великих обсягів відео та особистих даних.
Технології AI	Використовує штучний інтелект для аналізу поведінки, автоматичне виявлення підозрілої активності.	Автоматизація через Monitor, але більше залежить від правил тестування, ніж від складного аналізу AI.	Поєднання автоматизації та участі проктора, аналіз підозрілих дій здійснюється як проктором, так і алгоритмами.	Використовує AI для автоматичного виявлення порушень поряд із живим моніторингом.
Можливості кастомізації	Високий рівень кастомізації налаштувань для різних рівнів безпеки.	Можливість налаштування політик для тестування, але з обмеженим інтерфейсом.	Гнучкі налаштування залежно від потреб організації чи інструктора.	Мінімальні опції налаштувань, основний акцент на простоті використання.
Глобальне використання	Підтримує багатомовні інтерфейси та адаптацію до різних країн.	Використовується переважно у Північній Америці, обмежена адаптація до інших ринків.	Широко використовується у США, але доступна для користувачів у всьому світі.	Глобально розповсюджена платформа з підтримкою різних часових зон.
Переваги	Висока масштабованість, низькі технічні вимоги, можливість автоматизації процесу.	Блокування доступу до сторонніх ресурсів через LockDown Browser, простота використання.	Інтеграція живого проктерингу з автоматизованими системами, гнучкість у налаштуваннях.	Простота використання, високий рівень довіри до системи завдяки участі кваліфікованих проктерів.
Недоліки	Висока залежність від інтернет-з'єднання, можливі питання конфіденційності.	Складності з інтеграцією складних AI-рішень, відсутність моніторингу поза екраном.	Можливі технічні проблеми з відеоспостереженням та приватністю даних.	Високі технічні вимоги, можливість стресу у користувачів через постійний контроль.

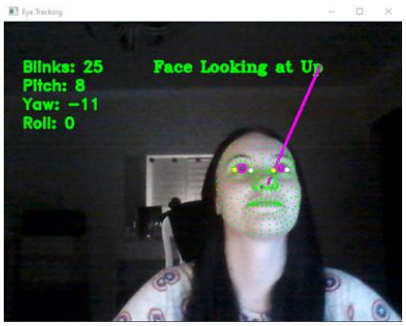
Модель алгоритму автоматизованого прийняття рішень на основі аналізу поведінки користувача

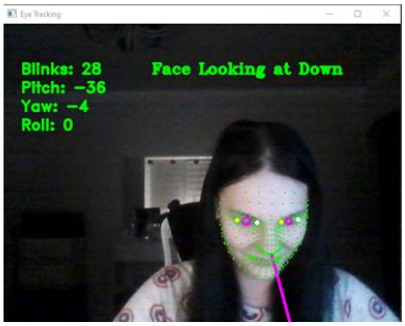


Тестування реалізованої системи

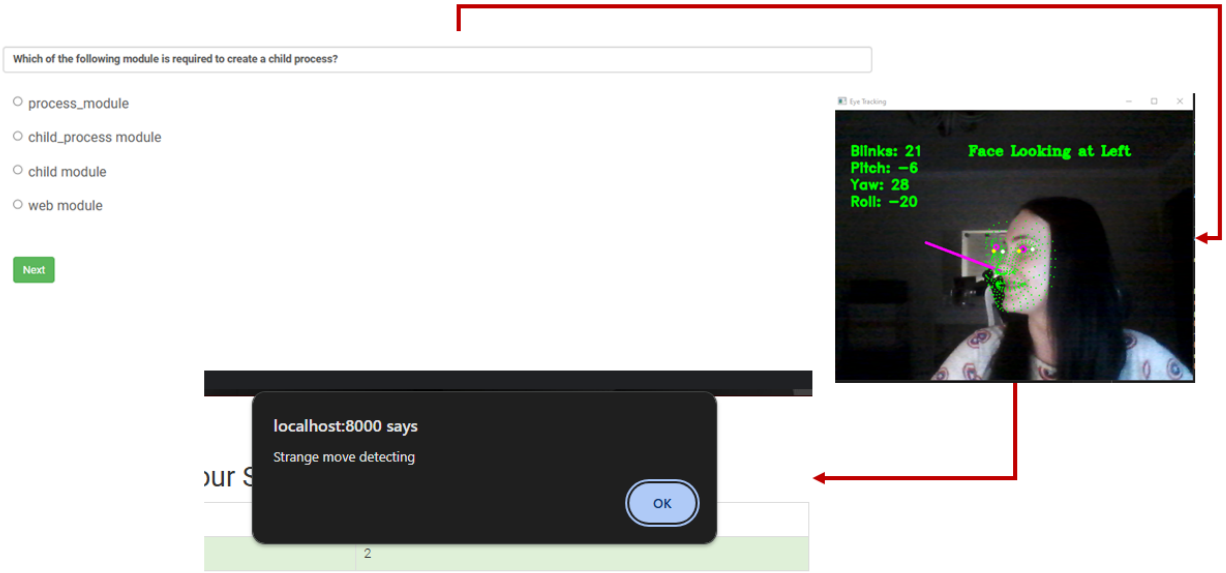


Тестування реалізованої системи

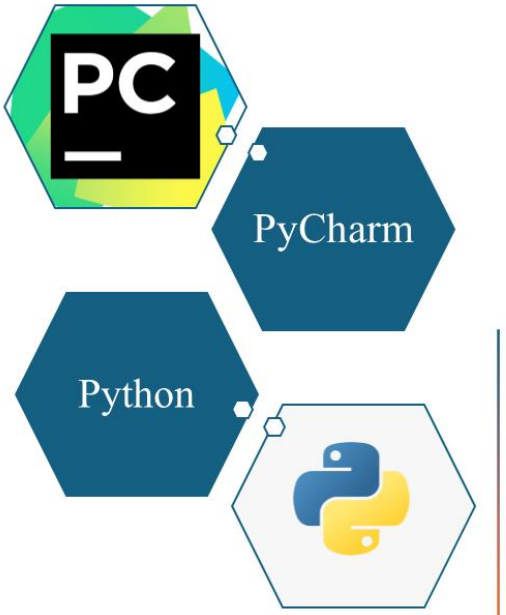




Тестування реалізованої системи



Використані технології



Висновок

Впровадження вдосконаленої системи оцінювання знань із використанням технологій відслідковування рухів і погляду не тільки забезпечує високий рівень контролю за діями користувача, але й відповідає сучасним вимогам у сфері кібербезпеки. Така система мінімізує ризики несанкціонованої поведінки під час тестування, підвищує об'єктивність оцінювання та сприяє захисту персональних даних.

Інтеграція цих технологій у процес оцінювання знань створює умови для прозорості, чесності та підвищує довіру до системи освіти. Це доводить актуальність та перспективність розробки в умовах швидкої цифровізації освітніх процесів і зростання кіберзагроз.

Дякую за увагу!

Додаток Г. Протокол перевірки на антиплагіат

ПРОТОКОЛ ПЕРЕВІРКИ НАВЧАЛЬНОЇ (КВАЛІФІКАЦІЙНОЇ) РОБОТИ

Назва роботи:

Вдосконалена система оцінювання знань на основі вдосконаленої моделі контролювання дій користувача з використанням технології відслідковування рухів та погляду користувача

Тип роботи: магістерська кваліфікаційна робота

Підрозділ Кафедра менеджменту на безпеки інформаційних систем
Факультет менеджменту та інформаційної безпеки
Гр. КІТС-23м

Керівник доцент Салієва О.В.

Показники звіту подібності

Turnitin	
Оригінальність	95%
Загальна схожість	5%

Аналіз звіту подібності (відмітити потрібне)

- ☐ Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- ☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її автора. Роботу направити на доопрацювання.
- ☐ 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.

Заявляю, що ознайомлений (-на) з повним звітом подібності, який був згенерований Системою щодо роботи (додається)

Автор _____ Гладка В.
(підпис) (прізвище, ініціали)

Опис прийнятого рішення

- ☐ **Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.**

Особа, відповідальна за перевірку _____ Коваль Н.П.
(підпис) (прізвище, ініціали)

Експерт _____ _____
(за потреби) (підпис) (прізвище, ініціали, посада)