

**ХУДОБА БОГДАН ПЕТРОВИЧ**

Кваліфікаційна наукова  
праця на правах рукопису  
УДК 004.652

**Методи та засоби аналізу параметрів оператора з  
використанням комп’ютерних тренажерів**

122 – комп’ютерні науки

**Дисертація на здобуття наукового ступеня  
доктора філософії**

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

\_\_\_\_\_ /Б.П. Худоба/

Науковий керівник

Камінський Роман Миколайович

доктор технічних наук, професор

## АНОТАЦІЯ

*Худоба Б.П.* Методи та засоби аналізу параметрів оператора з використанням комп'ютерних тренажерів. – Кваліфікаційна наукова праця на правах рукопису. Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 122 “Комп'ютерні науки”. – Національний університет «Львівська політехніка», Львів, 2023.

*Зміст анотації.* Дисертація присвячена побудові методів та засобів для аналізу параметрів оператора з використанням комп'ютерних тренажерів. Основна частина роботи включає вступ, чотири розділи та висновки.

У вступі розглянуто актуальність теми, поставлено мету та наукові задачі виділено наукову новизну та практичну цінність. Для досягнення мети дисертаційної роботи поставлено 6 цілей, що виконується по чергово.

Існує значна кількість автоматизованих систем дистанційного управління різноманітними об'єктами. Такі системи вимагають наявності кваліфікованих операторів для здійснення відповідних завдань у різних умовах. Наприклад, в побутовому середовищі, такому як забезпечення теплом у багатоквартирних будинках або житлових комплексах, управління процесом теплопередачі також потребує висококваліфікованих фахівців. Вони, відповідно до інформації, яку отримують на екрані монітора або інформаційному щиті, приймають відповідні рішення.

Іншим прикладом є авіадиспетчер, фахівець, який відслідковує дані з авіаційного радару, контролює вибір літаками відповідної смуги для посадки та зльоту, а також визначає наявність сторонніх об'єктів на їхньому шляху. Дуже важливою в сучасному світі є роль операторів безпілотних літальних апаратів (БПЛА), які стикаються з багатьма динамічними ситуаціями, що вимагають негайних та рішучих дій.

У цьому контексті професія оператора є надзвичайно важливою з точки зору підготовки відповідних фахівців у відповідний час. Отримання практичних навичок та первинного досвіду для таких професій забезпечується за допомогою різних типів комп'ютерних тренажерів. Таким чином, відбір, навчання та

атестація фахівців операторських професій є актуальною науково-технічною проблемою.

Аналіз літературних джерел свідчить про значний інтерес до вирішення такого типу завдань, зокрема, щодо підготовки операторського персоналу. Наприклад, моделювання продуктивності оператора в автономних системах здійснюється на основі ймовірнісного підходу та рекурентних нейронних мереж. Проте, в таких дослідженнях не враховується вплив фактору стресу на якість виконуваних оператором завдань.

У першому розділі «Аналітичний огляд літературних та інших джерел» проаналізовано поняття комп'ютерного тренажеру та людини-оператора. Розглянуто існуючі методи та засоби для аналізу параметрів оператора. Проаналізовано використання методу «людина в циклі». На основі аналізу літературних джерел було проведено систематизацію параметрів оператора, що піддаються оцінці за допомогою відомих методів. Виділено параметри, доступні для збору під час взаємодії оператора з розумними пристроями. Детально проаналізовано концепцію стресостійкості та запропоновано методи визначення параметра адаптивності оператора.

Досліджено використання комп'ютерних тренажерів для проведення тестів під час роботи з інформаційно-пошуковими системами. Висвітлено особливості конструювання тестів, зокрема їх структурування за рівнем складності та можливість трансформації оригінальних зображень для покращення ефективності тестування. Сформульовано завдання для подальшого наукового дослідження у межах дисертаційної роботи.

У другому розділі «Методи та засоби моделювання операторської діяльності» розглянуто методи та засоби аналізу параметрів оператора. Було проведено аналіз параметрів оператора комп'ютерного тренажеру та їх впливу на роботу, точність і ефективність процесу. Вибрано ключові параметри для подальшого дослідження їхнього впливу на результативність.

Детально проаналізовано математичну модель та концепцію тесту для комп'ютерного тренажеру, ідентифікувано етап пошуку об'єкта, етап складності його ідентифікації та етап обробки об'єкту пошуку. У цьому розділі розроблено

комп'ютерний тренажер для аналізу параметрів оператора, представлено його основні характеристики, функціональні можливості та результати експерименту.

Розглянуто східцеву функцію як процес роботи з тестом, визначено моменти часу  $t$  для побудови функції та розгортки періодичної функції. Побудовано східцеву функцію та зміщену обернену для ідентифікації моментів стресу та імпульсів відпочинку оператора під час виконання тренажеру.

У третьому розділі «Розробка методів моделювання операторської діяльності» було розроблено методи комп'ютерний тренажер моделювання операторської діяльності. Проведено кластеризацію операторів за їхніми параметрами на рівні загального та рівнів деталізації. Використано параметри, такі як фрактальна розмірність та показник Герста, для класифікації операторів.

Здійснено ранжування операторів в межах кластерів за допомогою методу багатовимірної середньої, що сприяло ефективному групуванню операторів та прогнозуванню їхньої ефективності. Проведено пошук аномалій у наборі даних та побудовано гістограми їх розподілу. Побудовано графіки залежності від різних параметрів оператора.

У даному розділі також розроблено метод-ансамбль класифікації для аналізу параметрів оператора за допомогою комп'ютерних тренажерів. Метод-ансамбль був створений шляхом навчання за допомогою методів класифікації даних. Точність отриманих результатів з методів класифікації перевищувала 0.85, проте подальше групування на кластери показало можливості підвищення точності на дані кластерів.

Представлено дерева рішень для загального набору даних, які відображають розподіл даних та вказують на основні властивості, що сприяють підвищенню точності результатів. Також наведено дерева рішень для окремих кластерів, де дані більш однорідні. Отримані результати навчання перевірено на наявність перенавчання.

У четвертому розділі «Розробка архітектури та апробація результатів» було проведено проектування сховища даних, призначеного для зберігання параметрів оператора для подальшого аналізу, а також технічних даних, необхідних для отримання інформації з розумних пристроїв. Додатково в цьому розділі була

представлена діаграма прецедентів, яка ілюструє основні сценарії використання системи комп'ютерного тренажеру та розподіл завдань між оператором та адміністратором. Також була розроблена діаграма безсерверної архітектури для розгортання системи в хмарному сервісі Amazon Web Services, обґрунтовано вибір цієї архітектури для проектованої системи.

У процесі проектування був включений механізм для зберігання файлів, необхідних для навчання та перевірки точності на нових даних, які зберігаються за днями та автоматично архівуються в кінці кожного робочого дня. Також була розроблена діаграма діяльності, що описує взаємодію між оператором, адміністратором та системою, а також діаграма станів, яка ілюструє процес проходження одного тесту оператором. Враховуючи типовий характер кожного тесту, діаграма станів відображає циклічний процес роботи оператора.

Деякі особливості технологічної реалізації системи були обґрунтовані, враховуючи особливості клієнтської частини (веб-застосунку), серверної частини (веб-сервера) та компоненту аналізу даних. Крім того, був наданий опис основних API для взаємодії між клієнтом та сервером.

Був розроблений користувацький інтерфейс, що ілюструється за допомогою скріншотів, демонструючи етапи проходження тестування та перегляду результатів для оператора. Після аналізу вимог до системи та отриманих результатів можна зробити висновок, що основні цілі та вимоги були досягнуті у процесі створення системи.

У висновка йдеться про результати, що були отримані в ході дослідження, де вони можуть використані та як до них дійшли.

Список літературних джерел наведений після висновків, що включає 91 літературне джерело.

У додатку А «Програмний код реалізації кластеризації та ієрархічного класифікатора» представлено код для виконання задачі кластеризації параметрів оператора та побудови ієрархічного класифікатора.

У додатку Б представлено таблиці Б.1. «Часові результати проходження тестів» та Б.2. «Результати проходження тестів». У цих таблицях наведено дані про 50 операторів, що брали участь у експерименті, а саме проходження 40 тестів

за комп'ютерним тренажером три рази. Таблиці містять описову статистику часу роботи кожного оператора та дані про успішність кожного оператора.

У додатку В представлено акти впровадження результатів дисертаційної роботи. Запропоновані методи і моделі впроваджені у навчальний процес Національного університету «Львівська політехніка» при викладанні дисципліни «Інтелектуальний аналіз даних» (підтверджено актом впровадження). Також результати дисертаційної роботи впроваджено в ПП «Скіфи» (підтверджено актом впровадження). Впровадження в ДБ «Інформаційна технологія формування психофізичного портрету в умовах стресових ситуацій» (№ держ. реєстру 0119U002257).

Основні наукові результати дисертації опубліковано в 7 працях, зокрема: три статті – у наукових фахових періодичних виданнях України; дві – у закордонних фахових періодичних виданнях; три публікації – у матеріалах міжнародних та всеукраїнських наукових, науково-технічних конференціях.

*Ключові слова:* людина-оператор, параметри оператора, комп'ютерний тренажер, методи класифікації та кластеризації, ієрархічний класифікатор, дендрограма, безсерверна архітектура.

## ABSTRACT

*Khudoba B.* Methods and means of analyzing operator parameters using computer simulators. – Qualifying scientific work on manuscript rights. Dissertation for obtaining the scientific degree of Doctor of Philosophy in specialty 122 «Computer Sciences». - Lviv Polytechnic National University, Lviv, 2023.

*Abstract content.* The dissertation is devoted to the construction of methods and tools for analyzing operator parameters using computer simulators. The main part of the work includes an introduction, four chapters and conclusions.

In the introduction, the relevance of the topic is considered, the goal and scientific tasks are set, scientific novelty and practical value are highlighted. To achieve the goal of the dissertation work, 6 goals are set, which are carried out alternately.

There is a significant number of automated remote control systems for various objects. Such systems require qualified operators to carry out the respective tasks in

various conditions. For example, in a domestic environment, such as providing heat in apartment buildings or residential complexes, the management of the heat transfer process also requires highly qualified specialists. They, in accordance with the information they receive on the monitor screen or information board, make appropriate decisions.

Another example is an air traffic controller, a specialist who monitors data from aviation radar, controls the choice of the appropriate lane for landing and takeoff by aircraft, and also determines the presence of foreign objects in their path. Very important in today's world is the role of unmanned aerial vehicle (UAV) operators, who are faced with many dynamic situations that require immediate and decisive actions.

In this context, the operator profession is extremely important from the point of view of training the appropriate specialists at the appropriate time. The acquisition of practical skills and initial experience for such professions is provided with the help of various types of computer simulators. Thus, the selection, training and certification of specialists in camera professions is an urgent scientific and technical problem.

The analysis of literary sources indicates a significant interest in solving this type of tasks, in particular, in relation to the training of camera personnel. For example, operator performance modeling in autonomous systems is based on a probabilistic approach and recurrent neural networks. However, such studies do not take into account the influence of the stress factor on the quality of tasks performed by the operator.

In the first chapter «Analytical review of literary and other sources» the concepts of computer simulator and human operator are analyzed. The existing methods and tools for analyzing operator parameters are considered. The use of the «human in the loop» method was analyzed. On the basis of the analysis of literary sources, a systematization of the parameters of the operator, which can be evaluated using known methods, was carried out. Parameters available for collection during operator interaction with smart devices are highlighted. The concept of stress resistance is analyzed in detail and methods for determining the operator's adaptability parameter are proposed.

The use of computer simulators for conducting tests while working with information and search systems has been studied. Features of test construction are highlighted, in particular, their structuring by level of complexity and the possibility of transforming original images to improve testing efficiency. Tasks for further scientific research within the framework of the dissertation work have been formulated.

In the second chapter «Methods and means of modeling operator activity» methods and means of analyzing operator parameters are considered. An analysis of the parameters of the computer simulator operator and their influence on the work, accuracy and efficiency of the process was carried out. Key parameters were selected for further investigation of their impact on performance.

The mathematical model and the concept of the test for the computer simulator were analyzed in detail, the stage of object search, the stage of its identification complexity and the stage of processing the search object were identified. In this section, a computer simulator for analyzing the parameters of the operator is developed, its main characteristics, functional capabilities and experimental results are presented.

The step function was considered as a process of working with the test, the moments of time  $t$  for the construction of the function and the sweep of the periodic function were determined. A step function and a shifted inverse were constructed to identify moments of stress and pulses of rest of the operator while performing the simulator.

In the third chapter, «Development of methods for simulating operator activity», computer simulator methods for simulating operator activity were developed. Operators were clustered according to their parameters at the general and detail levels. Parameters such as fractal dimension and Hirst exponent were used to classify the operators.

Operators were ranked within clusters using the multivariate average method, which contributed to the effective grouping of operators and forecasting of their efficiency. Anomalies in the data set were searched and histograms of their distribution were constructed. Graphs of dependence on various parameters of the operator are constructed.



In this section, an ensemble method of classification is also developed for the analysis of operator parameters with the help of computer simulators. The ensemble method was created by training using data classification techniques. The accuracy of the obtained results from the classification methods exceeded 0.85, but further grouping into clusters showed the possibility of increasing the accuracy of the cluster data.

Decision trees are presented for the common data set, which reflect the distribution of the data and indicate the main properties that contribute to the accuracy of the results. Decision trees for individual clusters where the data are more homogeneous are also given. The obtained training results were checked for retraining.

In the fourth chapter, «Architecture development and results validation», the design of a data warehouse designed to store operator parameters for further analysis, as well as technical data needed to obtain information from smart devices, was carried out. In addition, this section presented a case diagram that illustrates the main scenarios of using the computer simulator system and the division of tasks between the operator and the administrator. A serverless architecture diagram was also developed for the deployment of the system in the Amazon Web Services cloud service, and the choice of this architecture for the designed system was justified.

In the design process, a mechanism was included to store the files necessary for training and accuracy testing on new data, which are stored by day and automatically archived at the end of each working day. An activity diagram was also developed that describes the interaction between the operator, the administrator and the system, as well as a state diagram that illustrates the process of passing one test by the operator. Given the typical nature of each test, the state diagram displays the cyclic process of the operator.

Some features of the technological implementation of the system were substantiated, taking into account the features of the client part (web application), the server part (web server) and the data analysis component. In addition, a description of the main APIs for interaction between the client and the server was provided.

A user interface was developed, illustrated with screenshots, showing the stages of testing and viewing results for the operator. After analyzing the requirements for the

system and the obtained results, it can be concluded that the main goals and requirements were achieved in the process of creating the system.

The conclusion refers to the results obtained during the research, where they can be used and how they were arrived at.

The list of literary sources is given after the conclusions, which includes 91 literary sources.

Appendix A «Program code for implementing clustering and hierarchical classifier» presents the code for performing the task of clustering operator parameters and building a hierarchical classifier.

Appendix B contains Table Б.1 «Time measurements of test results» and Б.2. «Test results». These tables show data on 50 operators participating in the experiment, namely passing 40 tests on a computer simulator three times. The tables contain descriptive statistics of the working time of each operator and data on the success rate of each operator.

Appendix B presents the acts of implementation of the results of the dissertation work. The proposed methods and models are implemented in the educational process of the National University «Lviv Polytechnic» when teaching the discipline «Intellectual data analysis» (confirmed by the act of implementation). Also, the results of the dissertation were implemented in the «Скіфи» PE (confirmed by the act of implementation). Implementation in DB «Information technology for the formation of a psychophysical portrait under conditions of stressful situations» (state register number 0119U002257).

The main scientific results of the dissertation were published in 7 works, in particular: three articles – in scientific specialist periodicals of Ukraine; two – in foreign professional periodicals; three publications – in the materials of international and all-Ukrainian scientific, scientific and technical conferences.

*Keywords:* human operator, operator parameters, computer simulator, classification and clustering methods, hierarchical classifier, dendrogram, serverless architecture.

## СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ

### Статті у виданнях інших держав:

1. Nataliya Shakhovska, Roman Kaminskyi and **Bohdan Khudoba** "Experimental study and clustering of operating staff of search systems in the sense of stress resistance" Front. Big Data, 23 October 2023 Sec. Medicine and Public Health Volume 6 - 2023 <https://doi.org/10.3389/fdata.2023.1239017> **Q2**
2. Nataliya Shakhovska, Roman Kaminskyi, **Bohdan Khudoba**, Vladyslav Mykhailyshyn, Ihor Helzhynskyi "A Novel Methodology Analyzing the Influence of Micro-Stresses on Human-Centric Environments" Computation 2023, 11(11), 224; <https://doi.org/10.3390/computation11110224> **Q2**

### Статті у фахових виданнях України:

3. Камінський Р.М., Шаховська Н.Б., **Худоба Б.П.** Фрактальний аналіз моделей текстів різних стилів, поданих цілочисельними еквідистантними послідовностями кількості літер у словах. Вісник Хмельницького національного університету. Серія: Технічні науки. – 2021. – №2 (295) ст.
4. Бойчук А. Р., Камінський Р. М., Шаховська Н. Б., **Худоба Б. П.** Вплив кольору тла зображення-тесту на час виявлення людиною-оператором об'єкта, локалізованого на ньому, в системах комп'ютерного тренінгу // Вісник Хмельницького національного університету. Серія: Технічні науки. – 2021. – № 4 (299). – С. 18–25.
5. Камінський Р., Шаховська Н., **Худоба Б.** Експериментальне дослідження та групування операторського персоналу пошукових систем в сенсі стресостійкості // Вісник Хмельницького національного університету. Серія: Технічні науки. – 2022. – № 5 (313). ст. 42-51

### Матеріали конференцій:

6. **Худоба Б.** Аналіз роботи оператора з використанням комп'ютерних тренажерів // Multidisciplinary academic notes. Science research and

practice : abstracts of XV International scientific and practical conference (Madrid, Spain; April 19 – 22, 2022). – 2022. – С. 685–686

7. **Худоба Б.** Використання баз даних для часових рядів та метрик роботи оператора // Theoretical and applied aspects of the development of science: abstracts of XVIII International Scientific and Practical Conference (Bilbao, Spain; May 09 – 12, 2023). – 2022. – С. 503–504

## ЗМІСТ

<b>ВСТУП .....</b>	<b>15</b>
<b>РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД ЛІТЕРАТУРНИХ ТА ІНШИХ ДЖЕРЕЛ.....</b>	<b>19</b>
<b>1.1. Поняття комп'ютерний тренажер.....</b>	<b>19</b>
<b>1.2. Людина-оператор.....</b>	<b>22</b>
<b>1.3. Аналіз методів використання комп'ютерних тренажерів.....</b>	<b>22</b>
<b>1.4. Аналіз моделей тестування з використанням комп'ютерних     тренажерів .....</b>	<b>26</b>
1.4.1. Аналіз аналогу моделі людина в циклі .....	27
1.4.2. Аналіз поведінки оператора в системі управління енергетикою .....	30
<b>Висновки до першого розділу. ....</b>	<b>31</b>
<b>РОЗДІЛ 2. МЕТОДИ ТА ЗАСОБИ МОДЕЛЮВАННЯ ОПЕРАТОРСЬКОЇ ДІЯЛЬНОСТІ .....</b>	<b>32</b>
<b>2.1. Параметри людини-оператора .....</b>	<b>32</b>
<b>2.2. Вимоги щодо організації експерименту .....</b>	<b>37</b>
<b>2.3. Організація експерименту .....</b>	<b>44</b>
<b>2.4. Метод аналізу впливу потоку мікрострессорів на діяльність     оператора на основі східцевих функцій.....</b>	<b>49</b>
<b>Висновки до другого розділу .....</b>	<b>54</b>
<b>РОЗДІЛ 3. РОЗРОБКА МЕТОДІВ МОДЕЛЮВАННЯ ОПЕРАТОРСЬКОЇ ДІЯЛЬНОСТІ.....</b>	<b>55</b>
<b>3.1. Метод аналізу персональних даних оператора та результатів     проходження тестів.....</b>	<b>55</b>
3.1.1. Логістична регресія .....	57
3.1.2. Метод xgboost.....	60
3.1.3. Метод дерево рішень .....	61
<b>3.2. Кластеризація.....</b>	<b>65</b>
3.2.1. Метод k-середніх.....	65
3.2.2. Ієрархічний кластерний аналіз .....	69
<b>3.3. Аномальні значення та закономірності .....</b>	<b>74</b>

<b>3.4. Метод ієрархічної класифікації для визначення рівня стресостійкості</b>	<b>81</b>
<b>Висновки до третього розділу</b>	<b>89</b>
<b>РОЗДІЛ 4. РОЗРОБКА АРХІТЕКТУРИ ТА АПРОБАЦІЯ РЕЗУЛЬТАТІВ</b>	<b>90</b>
<b>4.1. Структура сховища даних</b>	<b>90</b>
<b>4.2. Архітектура системи</b>	<b>95</b>
<b>4.3. Розробка системи</b>	<b>102</b>
4.3.1. Розробка веб-застосунку	102
4.3.2. Розробка веб-сервера	104
4.3.3. Розробка методів аналізу даних	108
<b>4.4. Апробація результату</b>	<b>109</b>
<b>Висновки до четвертого розділу</b>	<b>114</b>
<b>ВИСНОВКИ</b>	<b>116</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ</b>	<b>118</b>
<b>ДОДАТОК А. ПРОГРАМНИЙ КОД РЕАЛІЗАЦІЇ КЛАСТЕРИЗАЦІЇ ТА ІЄРАРХІЧНОГО КЛАСИФІКАТОРА</b>	<b>127</b>
<b>ДОДАТОК Б. ТАБЛИЦІ РЕЗУЛЬТАТІВ ТЕСТУВАННЯ ОПЕРАТОРІВ...</b>	<b>128</b>
<b>ДОДАТОК В. АКТИ ВПРОВАДЖЕННЯ</b>	<b>132</b>

## ВСТУП

### **Актуальність теми.**

У даний час існує велика кількість автоматизованих систем дистанційного управління різними об'єктами. Такі автоматизовані системи потребують висококваліфікованих операторів для виконання відповідних робіт у різноманітних умовах. У побутових умовах, наприклад забезпечення теплом від центральної мережі багатоквартирного будинку або житлового комплексу, управління процесом теплопередачі також вимагає висококваліфікованих фахівців, які на основі інформації, отриманої на екрані монітора або на інформаційному щиті приймають відповідні рішення.

Інший приклад авіадиспетчер, спеціаліст, що виконує моніторинг карти чи даних з авіаційного радару, контролює вибір літаками відповідної смуги та безпеки посадки та зльоту, а також чи немає сторонніх об'єктів на їх шляху. Вельми важливою в наш час є роль операторів БПЛА, у них є багато динамічних ситуацій, що потребують негайних і рішучих дій.

У цьому плані професія оператора є надзвичайно важливою з точки зору підготовки відповідних фахівців в оптимальних відрізках часу. Навчання практичних навиків та набуття первинного досвіду для таких професій забезпечує використання різних типів комп'ютерних тренажерів. Тому відбір, навчання та атестація фахівців операторських професій є актуальною науково-технічною задачею. Аналіз літературних джерел показує значний інтерес до розв'язання такого типу задач – а саме підготовки операторського персоналу. Зокрема, моделювання продуктивності оператора в автономних системах здійснюється основі на ймовірнісного підходу, а також на основі рекурентних нейронних мереж. Проте, не враховується фактор стресу та його вплив на подальшу якість виконуваних оператором задач.

Важливим чинником з цієї точки зору, а саме підготовки оператора, є аналіз динаміки його параметрів в процесі виконання ним роботи. У даному дослідженні, у якості таких параметрів використано: частоту серцевих скорочень – пульс та час опрацювання наданого на моніторі тесту, що відображає фрагменти реальної робочої ситуації. Вибір частоти пульсу зумовлений тим, що

частота пульсу оператора пов'язана з концентрацією уваги в процесі сприйняття та пошуку об'єктів уваги на зображенні на моніторі робочої ситуації.

Отже, розроблення моделей, методів машинного навчання та засобів для підвищення якості оцінювання та класифікації параметрів оператора та експрес оцінки операторського персоналу є актуальною науково-практичною задачею.

**Зв'язок роботи з науковими програмами, планами і темами.** Дисертація виконувалася відповідно до пріоритетних напрямків науково-дослідних робіт Національного університету “Львівська політехніка”, відповідно до координаційних планів Міністерства освіти і науки України. Зокрема, в рамках наукових досліджень, виконуваних відповідно до держбюджетної роботи кафедри систем штучного інтелекту «Технології опрацювання мультимодальних україномовних даних для визначення рівня стресу» (№ держ. реєстру 0123U100231).

#### **Мета дисертаційного дослідження**

Метою дисертаційної роботи є розроблення моделей, методів машинного навчання та засобів для підвищення якості оцінювання та класифікації параметрів оператора та експрес оцінки операторського персоналу.

Для досягнення поставленої мети в роботі потрібно розв'язати такі задачі:

1. Провести аналіз існуючих інформаційних технологій в галузі моделювання стану людини-оператора та аналізу його параметрів;
2. Організувати комп'ютерний тренажер, забезпечений набором тестових завдань для симуляції пошукової діяльності людини-оператора;
3. На основі експериментальних досліджень провести аналіз впливу складності тестових завдань на його діяльність;
4. У результаті аналізу статистичних даних організувати систему оцінок операторської діяльності;
5. Розробити метод для оцінювання параметрів операторської діяльності;
6. Розробити інформаційну технологію для експрес оцінки параметрів операторської діяльності з використанням розглянутих методів.

**Об'єкт дослідження** – процеси сприйняття інформації людини-оператора наданої на екрані монітору.



**Предмет дослідження** – методи, модель людини-оператора та професійні тести, опрацювання даних оператора, організація процесу тестування з допомогою комп'ютерного тренажеру.

**Методи дослідження.** У процесі розробки моделі людино-машинного інтерфейсу в пошукових системах використано математичну статистику, теорію множин. Для побудови методу аналізу впливу потоку мікростресів на діяльність оператора використано модель східцевих функцій, теорію адгоритмів. Для побудови методу класифікації для визначення рівня стресостійкості використано моделі машинного навчання з вчителем та без вчителя, методи ансамблювання, кластерний аналіз та обробка багатовимірних даних.

### **Наукова новизна одержаних результатів**

- Уперше розроблено модель аналізу впливу потоку мікростресорів на діяльність оператора, яка відрізняється використанням східцевих функцій, що дає можливість підвищити достовірність моделювання уваги та часу прийняття рішення оператора за рахунок більш адекватного урахування стресових факторів.
- Удосконалено ієрархічний класифікатор визначення рівня стресостійкості оператора за рахунок попередньої кластеризації його параметрів та динамічного зважування виходів слабких класифікаторів ансамблю, що дало змогу підвищити індекс Джіні в батьківському вузлі та побудувати збалансоване дерево прийняття рішень при визначенні характеристик оператора.
- Отримав подальший розвиток метод аналізу діяльності людини-оператора в системах пошуку об'єктів заданого класу, який відрізняється від існуючих урахуванням під час кластеризації показника Герста та фрактальної розмірності, що дає можливість підвищити якість аналізу діяльності людини-оператора.

### **Практичне значення одержаних результатів**

Розроблено комп'ютерний тренажер і генератор тестових зображень. Вони дозволяють оцінити стресостійкість оператора. Генерується послідовність зображень в окремі моменти. На екрані монітора оператору надається послідовність тестових зображень з об'єктами уваги заданого класу. Значення пульсу і час опрацювання тесту є основними параметрами оператора. Моменти

їх впливу та рішення, прийняті оператором, фіксуються, а їх значення вносяться до протоколу дослідження і впринципі дають підстави для оцінки кваліфікації даного оператора.

Розроблено схему зберігання даних, архітектури системи та її основних модулів, обґрунтовано технології та засоби для реалізації системи. Представлено функціонал системи, основні сценарії роботи з користувачем та адміністратором.

Запропоновані методи і моделі впроваджені у навчальний процес Національного університету «Львівська політехніка» при викладанні дисципліни «Інтелектуальний аналіз даних» (підтверджено актом впровадження).

Також результати дисертаційної роботи впроваджено в ПП «Скіфи» (підтверджено актом впровадження).

Впровадження в ДБ «Інформаційна технологія формування психофізичного портрету в умовах стресових ситуацій» (№ держ. реєстру 0119U002257).

**Особистий внесок здобувача.** Основні положення та результати дисертаційної роботи одержані автором самостійно. Особисто здобувачеві належать наступні наукові результати: розроблено модель людино-машинного інтерфейсу в пошукових системах [1, 2], розроблено метод ієрархічної кластеризації [3], розроблено алгоритм генерації тестових зображень [4], розроблено метод кластеризації для визначення рівня стресостійкості [5].

**Апробація результатів дисертації.** У результаті дисертації розроблена і запроваджена інформаційна технологія застосування людино-машинного інтерфейсу в пошукових системах. Роботу апробовано на Міжнародній науково-практичній конференції «Мультидисциплінарні наукові записки» (2022), Міжнародній науково-практичній конференції «Теоретичні та прикладні аспекти розвитку в науці» (2023).

**Публікації.** За результатами виконаних досліджень опубліковано 7 наукових праць, із них 2 статті – у виданнях, що індексуються в наукометричних базах даних (дві статті в журналах з Q2), 3 статті – у фахових виданнях України, 2 публікації – у збірниках наукових праць конференцій.

**Структура та обсяг дисертації.** Дисертація складається із вступу, чотирьох розділів, висновків, списку використаних джерел із 91 найменування. Повний обсяг дисертації складає 134 сторінок, основний зміст викладено на 103 сторінках, де наведено 44 рисунки та 7 таблиць.

## РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД ЛІТЕРАТУРНИХ ТА ІНШИХ ДЖЕРЕЛ

У розділі здійснено опис комп'ютерного тренажеру, людини-оператора комп'ютерного тренажеру та можливості, які дає навчання на комп'ютерному тренажері. Наведено основні фактори, які впливають на роботу оператора. Здійснено опис експериментального тесту для людини-оператора, що використовується з метою дослідження його поведінки в різних умовах. Поставлено задачу проаналізувати як саме впливає реакція на стрес, здатність до прийняття рішень на роботу оператора.

### 1.1. Поняття комп'ютерний тренажер

Комп'ютерні тренажери – це інноваційний засіб навчання та тренування, який використовує віртуальні симуляції для створення імітації різних сценаріїв та ситуацій [1]. Вони дозволяють користувачам отримати практичний досвід та навички в безпечному та контрольованому середовищі. Комп'ютерні тренажери застосовуються в широкому спектрі галузей, від медицини та авіації до військових справ та освіти.

Комп'ютерні тренажери можуть бути фізичними пристроями або програмними додатками, які дозволяють користувачам взаємодіяти з віртуальним середовищем, що імітує реальну дійсність. Ці системи використовуються для тренування навичок, вдосконалення реакцій, вирішення проблемних завдань та відпрацювання критичних сценаріїв [2].

Це лише кілька прикладів типів комп'ютерних тренажерів. Вони також застосовуються в автомобільній промисловості для тренування водіїв, у виробництві для навчання робочих процесів та в освітній сфері для покращення навичок учнів.

Комп'ютерні тренажери поділяються на декілька видів відповідно до їхніх застосувань та особливостей. Розглянемо декілька головних видів:

- **Медичні тренажери.** Медичні комп'ютерні тренажери використовуються для навчання медичним працівникам і студентам. Наприклад, хірургічні

симулятори дозволяють вправлятися у хірургічних процедурах без ризику для пацієнтів. Також існують тренажери для навчання діагностиці та лікуванню різних захворювань.

- **Авіаційні тренажери.** Авіаційні комп'ютерні тренажери призначені для навчання льотному складу та пілотам. Вони дозволяють відтворювати реальні умови польоту, включаючи екстремальні ситуації і аварії.
- **Військові тренажери.** Військові комп'ютерні тренажери використовуються для навчання військовому персоналу різних видів військової діяльності, включаючи стрільбища, симуляцію бойових операцій та тактичні тренування.

Це лише кілька прикладів типів комп'ютерних тренажерів. Вони також застосовуються в автомобільній промисловості для тренування водіїв, у виробництві для навчання робочих процесів та в освітній сфері для покращення навичок учнів.

Використання комп'ютерних тренажерів має багато переваг, серед яких:

- **Безпека.** Комп'ютерні тренажери дозволяють проводити тренування та навчання в безпечних умовах, без реальних ризиків для життя та здоров'я.
- **Ефективність навчання.** Вони дозволяють збільшити ефективність навчання, оскільки можуть створювати реалістичні ситуації та навчальні середовища.
- **Зменшення витрат.** Використання комп'ютерних тренажерів може зменшити витрати на тренування та навчання, оскільки вони не потребують реальних об'єктів та ресурсів.
- **Моніторинг та оцінка.** Комп'ютерні тренажери дозволяють збирати дані про продуктивність користувачів та надавати об'єктивну оцінку їхніх навичок.
- **Адаптивність.** Деякі комп'ютерні тренажери можуть адаптуватися до рівня знань та навичок користувача, надаючи індивідуальні завдання та завдання.

- **Моделювання складних сценаріїв.** Комп'ютерні тренажери дозволяють створювати реалістичні симуляції складних сценаріїв, включаючи аварійні ситуації, які важко змоделювати в реальному житті.

Переважаючо комп'ютерні тренажери створені з метою навчання операторського персоналу. Оператор відіграє важливу роль в сучасних симуляційних системах, де використовуються комп'ютерні програми та обладнання для навчання, тренування, а також симуляції різних сценаріїв. Робота оператора є ключовою для забезпечення успішного використання комп'ютерних тренажерів у різних галузях, включаючи медицину, авіацію, військову справу, автомобільну індустрію та багато інших.

Оператор на навчальному тренажері вчиться виконувати роботу з метою покращити свої знання та навички перед роботою за справжнім пристроєм.

Виконуючи завдання на тренажері оператор може отримати розумове та фізичне навантаження різного рівня. Таким чином можна прослідкувати як оператор виконує різні періоди в роботі, від базового з повільним темпом до інтенсивного з вимогою приймати складні рішення. Враховуючи той факт, що розробник тренажеру складає різні плани роботи та навантаження на тренажері, можна скласти сценарії з різним рівнем навантаження, а також поєднувати їх. Це дає змогу прослідкувати як різні темпи впливають на оператора. Також кожен розроблений план має свій час роботи за тренажером.

Одним з важливих видів операторської діяльності є робота людини-оператора за комп'ютерним тренажером, що імітує роботу авіаційного радару [3]. Ця робота має суттєве значення для навчання та тренування диспетчерів повітряного руху та операторів радарів. Комп'ютерний тренажер імітує роботу реального радару, відтворюючи повітряний простір та рух літаків та інших повітряних об'єктів. Оператор може спостерігати їх рух та взаємодію відображених об'єктів. Також може використовувати комп'ютерний тренажер для визначення положення, швидкості, висоти та інших параметрів повітряних об'єктів, які відображаються на екрані радару. Завданням оператора на такому тренажері є вчасна ідентифікація об'єктів. Якщо виявлений об'єкт очікуваний то потрібно просто

слідкувати за ним або скерувати до відповідної служби. Якщо ж ні, то повідомляти про сторонній об'єкт у повітряному просторі.

## **1.2. Людина-оператор**

Людина-оператор на комп'ютерного тренажеру проходить шлях від навчання на ньому, виконуючи завдання різного рівня складності тестування. Етап тестування передбачає відслідковування різних параметрів самого оператора, а також успішність та швидкість виконання завдання. Такі завдання на відміну від навчання в школі чи університеті мають практичний характер та передбачають прямий зв'язок між типом і складністю в тесті та в реальному режимі роботи оператора пристрою, що використовується в житті [4]. Тому важливо розуміти, які параметри та як саме можна аналізувати та тестувати. За результатами можна сказати чи готова людина до праці чи потребує додаткового навчання.

У таких галузях як авіація, медицина, управління автомобілем це важливо зрозуміти перед початком роботи, бо від цього залежить життя людей. Проте після проходження навчання і тестування на комп'ютерному тренажері слід відслідкувати чи змінюються результати людини в реальному процесі, так як це залежить від психологічного і фізичного стану людини. Людина може як і покращувати, так і погіршувати результати порівняно з тренажером.

Людина-оператор через виконання своїх обов'язків може перебувати в стані стресу. Стрес можна класифікувати за тривалістю: актуальний стрес(мікрострес) – тимчасовий, короткотривалий стрес, який виникає в реакцію на конкретну ситуацію або подію та хронічний стрес – довготривалий стрес, який триває протягом тривалого часу і може бути пов'язаний з постійними труднощами або невирішеними проблемами.

## **1.3. Аналіз методів використання комп'ютерних тренажерів**

Розробленню комп'ютерних тренажерів присвячено різні дослідження. Зазвичай такі тренажери розробляються на основі методів аналізу параметрів оператора.

Модель перебування та виходу оператора зі стресу розглянуто в [5], а також пояснюється виникнення нервово-психічного перенапруження. Теоретичні основи психодіагностики стресу та методи, пов'язані з процедурою психологічної діагностики, етикою та етапами психолого-діагностичного обстеження наведені в [6]. У праці [7] автори розглядають раціональне використання симуляцій для підтримки авіаційного навчання. Вони обговорюють ролі комп'ютерних тренажерів у підготовці авіаторів та використання симуляційних засобів. Автор статті [8] досліджує майбутнє використання симуляцій у галузі охорони здоров'я, включаючи медичну симуляцію та комп'ютерні тренажери для навчання медичному персоналу. У роботі [9] розглядається використання ігор для навчання, включаючи серйозні ігри, які використовують комп'ютерні тренажери для навчання студентів та професіоналів у різних галузях. У статті [10] розглядаються особливості комп'ютерної підготовки операторів безперервних технологічних процесів у порівнянні з іншими предметними галузями. Робота [11] проводить систематичний огляд використання комп'ютерних симуляторів для командного навчання та оцінки в галузі охорони здоров'я. Ця стаття [12] досліджує вплив відеоігор на агресивні думки, почуття та поведінку. Вона вказує на те, що комп'ютерні тренажери, такі як відеоігри, можуть впливати на психологічний стан користувачів. Праця [13] проводить систематичний огляд концептуального каркасу комп'ютерного навчання в галузі охорони здоров'я, включаючи використання комп'ютерних тренажерів. Робота [14] автор досліджує вплив крос-тренування та навантаження на функціонування команд у симуляційних середовищах, включаючи комп'ютерні тренажери. У статті [15] розглядається використання комп'ютерних симуляторів для тренування навичок в оптиці та хірургії. Автори досліджують вплив комп'ютерних тренажерів на процеси та результати роботи команд у праці [16]. У роботі [17] автор розглядає практики співробітництва в створенні та використанні комп'ютерних тренажерів. Ці джерела відображають різні аспекти та галузі використання комп'ютерних тренажерів, включаючи навчання, медицину, авіацію та багато інших. Вони

надають уявлення про сучасні дослідження та застосування цієї технології в різних контекстах.

Книга [18] надає обширний огляд параметрів людини-оператора, таких як когнітивні, фізіологічні та психомоторні аспекти, які впливають на ефективність роботи з комп'ютерними тренажерами. У статті [19] автор досліджує поняття свідомості ситуації та параметри, що впливають на розуміння оператором динамічних систем, включаючи комп'ютерні тренажери. Ця стаття [20] розглядає психологічні та нейроергономічні аспекти параметрів оператора в контексті навчання за допомогою комп'ютерних тренажерів. Дана праця [21] досліджує ефективність різних когнітивно-поведінкових методів навчання операторів на комп'ютерних тренажерах. Стаття [22] вивчає вплив характеристик завдань, різниці між користувачами та перерв на ефективність роботи операторів з комп'ютерними тренажерами. Робота [23] розглядає концепцію множинних ресурсів та її роль у передбаченні продуктивності операторів на комп'ютерних тренажерах. У статті [24] наведено приклад розпізнавання реалістичного зображення з використанням як ручного, так і автоматичного тестування з таблицею рішень. Автор дослідження [25] описує про взаємодії пілотів з автоматизованими системами в літаках та їх вплив на параметри операторів. У статті [26] розглядається важливість параметрів, таких як свідомість ситуації, психологічне навантаження та довіра до автоматизованих систем у роботі оператора. Дослідження [27] взаємодії між параметрами контролю, довіри та розподілу функцій у системах людина-машинна на прикладі комп'ютерних тренажерів. Стаття [28] оглядає методи вимірювання свідомості ситуації, як важливого параметру для роботи операторів з комп'ютерними тренажерами. Ці джерела допомагають глибше зрозуміти параметри, які впливають на роботу оператора комп'ютерного тренажеру і їх важливість в психологічному та інженерному контекстах.

Стаття [29] досліджує поняття свідомості ситуації, психологічне навантаження та довіру до автоматизованих систем у роботі операторів комп'ютерних тренажерів. Автор книги [30] надає методи і підходи до врахування факторів людських факторів у дизайні та інженерії комп'ютерних тренажерів.



Праця [31] аргументує важливість психологічних аспектів в інженерії та дизайні комп'ютерних тренажерів. Книга [32] надає огляд психологічних аспектів впливу на продуктивність операторів комп'ютерних тренажерів. Ці джерела дають можливість зрозуміти ключові аспекти роботи людини-оператора на комп'ютерних тренажерах і важливість психологічних та інженерних параметрів для підвищення ефективності навчання та тренування.

Наступні джерела нададуть розуміння важливості вивчення впливу стресу на операторів комп'ютерних тренажерів і методів зменшення негативних наслідків стресу для підвищення продуктивності та безпеки роботи. Дослідження [33] присвячена розробці та застосуванню індексу навантаження завдання (NASA-TLX), який допомагає оцінити вплив стресу та психологічного навантаження на оператора комп'ютерного тренажеру. Автор статті [34] досліджує вплив стресу на мозок, поведінку та когніцію на протязі життя, що може бути важливим для роботи оператора на комп'ютерних тренажерах на різних етапах життя. Стаття [35] розповідає, чи призводить стрес до втрати командної перспективи у роботі операторів комп'ютерних тренажерів. Автор праці [36] говорить про вплив фізіологічних змін, таких як менструальний цикл, на настрій та когнітивну продуктивність операторів комп'ютерних тренажерів. Розділ книги [37] досліджує, як дизайн систем та інтерфейсів може впливати на сприйняття та управління стресом операторами комп'ютерних тренажерів. Стаття [38] розглядає психосоціальні фактори, які впливають на стрес та робочу продуктивність операторів на комп'ютерних тренажерах.

Результати аналізу літературних джерел та їх вплив на вирішення задачі розроблення комп'ютерних тренажерів подано у таблиці 1.1.

Таблиця 1.1. Результат огляду літератури.

Автори	Основний внесок	Значення
Dykstra, J., & Paul, C. L. [11]	Порівняння стресових підходів кібероперацій	Робота описує вплив стресу на працівника кібербезпеки. Вивчення впливу фрустрації, помилок на роботу співробітника, який

		використовує комп'ютерний тренажер або просто комп'ютер, допомагає зрозуміти різні стани, причини і наслідки.
H. Zhu, D. Liu, I. Bayley, R. Harrison, and F. Cuzzolin [24]	Метод тестування інтелектуальних програм.	Тестування інтелектуальних програм дає зрозуміти, як вручну та автоматично розпізнавати певні елементи. Підходи, що використовуються у автоматизованих системах.
Barak, O., & Tsodyks, M [6]	Математичні моделі навчання в нейробіології	У статті описано математичні моделі та їх використання в нейробіології. Також методи застосування аналізу даних для вивчення поведінки операторів симулятора.
Şahin, C., Rokne, J., & Alhajj, R. [5]	Математична модель поведінки людини для імітації евакуації будівель під час надзвичайних ситуацій	Автор розповідає про моделювання поведінки людей в надзвичайних ситуаціях. Тому моделювання таких ситуацій пов'язане з моделюванням мікростресової ситуації під час роботи оператора.

Отже, методи розроблення комп'ютерних тренажерів є дослідженими, проте приділено мало уваги визначенню параметрів, які впливають на вхід та вихід оператора з мікростресу, а також на ефективність його подальшої роботи.

#### **1.4. Аналіз моделей тестування з використанням комп'ютерних тренажерів**

Проаналізуємо існуючі моделі тестування з використанням комп'ютерних тренажерів.

### 1.4.1. Аналіз аналогу моделі людина в циклі

Розглянемо роботу [39] про продуктивність оператора в системі типу людина-в-циклі (human-in-the-loop). Автор розглядає важливість включення людини-оператора в напів-автономних та автономних системи. Автоматизація – це прогрес, що допомагає пришвидшити процеси, уникнути рутинної роботи у багатьох галузях, але зазначається, що вона не завжди є бажана через можливі негативні наслідки для операцій системи. Існують випадки коли потрібно враховувати складність і нюанси людської експертизи та втручання, семі-автономні системи та співробітництво між людьми та машинами є більш важливими для досягнення найкращих результатів.

Далі автором розглядається продуктивність системи та продуктивність людини-оператора. Існують фактори, такі як тип завдання, поточне навантаження та інші змінні середовища, що можуть впливати на продуктивність системи. Також продуктивність системи залежить від характеристик оператора, включаючи довгострокові та короткострокові особливості. Моделювання оператора є корисним для розуміння продуктивності системи та підтримки конкурентної переваги підприємства. Моделювання допомагає виявити недоліки в системі, розробити налаштовані мотиваційні стратегії та відібрати відповідних кандидатів для досягнення цілей системи. Формальні методи та формальна верифікація дозволяють математично аналізувати можливі поведінки дизайну системи та забезпечують надійність результатів. Завдяки формальним методам можна отримати гарантії правильності дизайну відповідно до визначених специфікацій. Система, яка є стохастичною, може бути моделювана за допомогою MDP. MDP використовується для аналізу системи та її поведінки для заданих специфікацій.

Наступним кроком надається коротка інформація щодо імовірнісного моделювання перевірки, моделі Маркова прийняття рішень (MDP), тимчасової логіки імовірнісного розширення та інтродукція до відомого моделювальника імовірнісних моделей PRISM. PRISM – це засіб перевірки імовірнісної моделі, інструмент для формального моделювання та аналізу систем, які демонструють випадкову або ймовірнісну поведінку. Модель MDP використовується для

захоплення поведінки системи в реактивних системах та базується на станах системи та їх ймовірній еволюції через вибір дій. Далі пояснюється використання логіки тимчасових властивостей для специфікації властивостей системи, інтродукція до PRISM як моделювальника імовірнісних моделей.

У цій статті пропонується новий підхід до моделювання продуктивності оператора в автономних системах НІТЛ. Цей підхід базується на ймовірнісній моделі, яка враховує різні фактори, що впливають на продуктивність оператора, наприклад профіль оператора, профіль робочого навантаження та політику перерв. Автори оцінюють запропонований підхід до сценарію, місії, спостереження БПЛА та показують, що він може точно прогнозувати роботу оператора за різних умов.

На продуктивність оператора впливає безліч факторів, зокрема вік, досвід, тип оператора, режим сну, умови робочого навантаження, тип завдання та втома.

Запропонована ймовірнісна модель може точно прогнозувати продуктивність оператора за різних умов, включаючи різні профілі операторів, профілі робочого навантаження та політики перерв.

Запропонована модель може бути використана для розробки кращих автономних систем НІТЛ шляхом оптимізації параметрів системи для максимізації продуктивності оператора.

Стаття робить значний внесок у сферу автономних систем НІТЛ. Запропонована модель може бути використана для підвищення продуктивності та безпеки автономних систем НІТЛ у різноманітних додатках, таких як спостереження БПЛА, навігація наземних транспортних засобів та співпраця людини та робота.

Представлений аналіз призначений для різних областей застосування, де оператор надає системі напіваавтономної допомоги. Модель розроблена так, щоб оцінювати різні місії, але необхідно враховувати, що в деяких застосуваннях взаємодія між людиною та автоматизованими компонентами може бути більш складною і не може бути модулярною. Для того, щоб модель була придатною в різних галузях, потрібно мати значення параметрів домену. Важливо здійснювати

дослідження в конкретній галузі, щоб отримати значення параметрів, які відображають її особливості.

Автор роботи провів моделював продуктивність роботи людини-оператора, беручи до уваги загальні параметри оператора, такі як вік, стиль сну, завантаженість. Такі параметри здебільшого є наперед відомі, їх можна використовувати та робити певний аналіз на їх основі, але ми не знаємо як міняється стан людини-оператора під час самого виконання завдання. Відслідковування таких параметрів як пульс, час відповіді, метеорологічні дані дають кращу картину розуміння стану оператора та є більш передбачуваними. Послідовність станів у ланцюгах Маркова показує як людина-оператор переходить від одного до іншого стану. Проте аналіз параметрів оператора не є повністю розкритий. Врахувавши такі параметри як пульс, час виконання, метеорологічні дані, вік, фізичний стан ми отримуємо повнішу картину. На основі таких даних можна зробити кластеризацію, поділивши операторів на різні кластери за успішністю, продуктивністю.

Можливість побудови комп'ютерного тренажеру та навчання на тренажері дасть змогу провести аналіз більшої кількості параметрів та отримати результати, які можна використовувати як систему рекомендацій та навчання операторів перед роботою на справжніх об'єктах.

Аналіз параметрів людини-оператора є важливим в багатьох галузях. Приклади з попередніх статей це оператор БПЛА, оператор управління енергосистеми, але це лише початок. Авіадиспетчер, оператор на підприємстві, пілот, машиніст потягу – ці всі професії та багато інших можуть вважатися операторами, вони виконують роботу, яку можна змоделювати на комп'ютерному тренажері та проводити навчання на ньому. Ба більше, часто так і відбувається, але не аналізуються параметри самого оператора та що впливає на його точність, швидкодію.

Завдяки такому розумінню можна уникнути авіакатастроф, помилок на будівництві, покращити роботу операторів БПЛА. Розуміння ситуацій, коли людина виконує добре роботу, а коли погано дає змогу вибрати кращі режими

чергування, відібрати кращих працівників, що підходять для певної професії, а це як мінімум економічні збереження. Також це може рятувати людські життя.

#### **1.4.2. Аналіз поведінки оператора в системі управління енергетикою**

У статті [40] пропонується основа для включення людської поведінки в системи управління енергосистемою. Структура базується на експерименті системи живлення людини в циклі (НІТЛ), який використовує оптимальний контролер як еталонну модель навчання для людей-операторів. Автори стверджують, що оптимальний контролер можна розглядати як ідеальну людину-оператора (тобто ідеально навченого оператора без несприятливих ефектів нервово-м'язової активації). Навчаючи людей-операторів працювати як оптимальний контролер, автори вважають, що ймовірність і вплив затримок і помилок оператора можна зменшити.

У статті представлені результати пілотного дослідження, в якому люди порівнювалися з оптимальним контролером на простій робочій станції оператора енергосистеми. Результати показують, що люди-оператори не здатні працювати так само добре, як оптимальний контролер, але вони можуть навчитися покращувати свою продуктивність за допомогою навчання. Автори дійшли висновку, що запропонована структура має потенціал для використання для підвищення безпеки та надійності енергетичних систем шляхом навчання людей-операторів роботі, схожій на оптимальний контролер.

Автори виявили, що оптимальний контролер здатний стабілізувати систему та мінімізувати середньоквадратичне значення вартості частотної характеристики ефективніше, ніж люди-оператори. Однак вони також відзначають, що оптимальний контролер представляє собою ідеальний випадок, який не може бути досягнутий людьми-операторами через такі фактори, як шум спостереження, моторний шум і нейромоторне відставання. Також відзначається, що люди-оператори мають різні стилі керування, причому одні оператори швидко й часто налаштовують вхідні дані системи, тоді як інші налаштовують їх більш плавно. Вони припускають, що ці відмінності можуть бути зумовлені такими

факторами, як знання предметної галузі, усвідомлення ситуації, розумове навантаження та досвід.

У роботі автори шукають оптимальне значення для роботи оператора по управлінню енергосистемою. Такі параметри не включають параметрів самого оператора, а більше параметри його роботи, що він має зробити і як. Не включається вплив фізичних чи емоційних параметрів. Автори стверджують, що людина-оператор не здатна досягти такої оптимальності як автоматизована система, але в свою чергу оператор потрібен, бо автоматизована система може не знати як діяти у нештатних ситуаціях чи коли потрібно приймати додаткові рішення, що не закладені в початковій умові. Тому досягнення такого оптимального значення як може досягти автоматизована система людиною-оператором – не можливо. Тому має сенс дослідження параметрів самого оператора та розуміння, що впливає на його швидкодію та успішність в роботі.

### **Висновки до першого розділу.**

У розділі проаналізовано літературні джерела [1-40]. Було розглянуто поняття комп'ютерних тренажерів та можливості їх застосування у різних сферах. Виділено роль людини-оператора тренажеру.

На основі літературних джерел зібрано інформацію про параметри оператора, що аналізувалися відомими методами та виділено параметри, які можна зібрати при роботі оператора з розумним пристроєм. Розглянуто поняття стресостійкості та як можна визначити параметр адаптивності оператора.

Розглянуто поняття тесту при роботі за комп'ютерним тренажером. Особливості в побудові тесту для інформаційно-пошукових систем, подання тестів різної складності та як їх можна трансформувати з оригінальних зображень.

Було поставлено задачі для продовження дисертаційного дослідження.

## РОЗДІЛ 2. МЕТОДИ ТА ЗАСОБИ МОДЕЛЮВАННЯ ОПЕРАТОРСЬКОЇ ДІЯЛЬНОСТІ

У розділі розглянуто різні параметри оператора, що можуть впливати на його роботу та ефективність. Організовано тренажер та описано експеримент по роботі з ним. Також наведено модель тесту з використанням комп'ютерного тренажеру. Наведено можливості використання східцевої функції для подання діяльності оператора під час навчання в системах візуального пошуку.

### 2.1. Параметри людини-оператора

Параметри людини-оператора – це характеристики та особливості, які визначають його здатність та ефективність виконання конкретних завдань у певному середовищі. Ці параметри відіграють важливу роль у різних професійних галузях, включаючи авіацію, медицину, військовий сектор та багато інших [41]. Найбільше використовують такі параметри оператора:

- **Стресостійкість.** Оператор повинен бути здатний ефективно працювати під час стресових ситуацій та зберігати холонокров'я.
- **Адаптивність.** Здатність адаптуватися до мінливих умов і непередбачених обставин є цінним параметром. Оцінка здатності оператора до адаптації допомагає переконатися, що він може працювати в різних умовах.
- **Фізичний стан.** Деякі завдання вимагають фізичних навичок, таких як координація рухів, дотримання точності та витривалість.
- **Спроможність приймати рішення.** Оператор повинен вміти швидко та правильно приймати рішення в надзвичайних ситуаціях.
- **Координація та реакція.** Оператори в сферах, де важливі реакція та координація, повинні володіти швидкістю реакції та точністю координації рухів.
- **Здатність до самоконтролю.** Оператор повинен бути здатний керувати своєю діяльністю та вживати заходи для запобігання помилкам.
- **Етика та відповідальність.** Деякі професії вимагають високих етичних стандартів та відповідальності за свої дії.



- **Когнітивні навички.** Оператор повинен мати високий рівень когнітивних навичок, включаючи здатність приймати рішення, вирішувати завдання та аналізувати інформацію.
- **Комунікаційні навички.** Важливо мати здатність чітко та ефективно комунікувати з іншими операторами, командою чи керівництвом.

Розглянуті параметри можуть варіюватися залежно від конкретної професії та завдань оператора. Важливо також їх виявляти, оцінювати та розвивати для забезпечення ефективності та безпеки в професійних сферах, де діють оператори.

Деякі з цих параметрів не мають чіткого визначення чи оцінки, але є ті що мають більшу вагу та можна оцінити безпосередньо або ж за допомогою інших параметрів.

Фізичний стан людини залежить від віку, наявності хронічних захворювань, точності зору. Адаптивність оцінюється, даючи оператору завдання різного рівня складності чи міняючи умови проходження завдання. Стресостійкість можна вимірювати зміною серцевого ритму в різних ситуаціях. Також на стресостійкість впливає соціальна ситуація в суспільстві [42], вміння приймати рішення і зміна серцевого ритму під час виконання оператором завдання. Протяжність роботи за тренажером також показує зміни в результатах роботи за тренажером. Оператор працюючи 30 хвилин виконує роботу швидше і точніше ніж той же оператор, що пропрацював уже 8 годин. Тому час проведений за тренажером теж слід брати до уваги.

Стрес і стресостійкість мають особливо важливу роль, так як людина у стані стресу може виконувати роботу зовсім інакше ніж у звичайному стані спокою. Є випадки, коли результат під час стресу покращується, але частіше таки зворотній ефект. Стрес і серцевий ритм мають глибоку і взаємозалежну зв'язок, і їх взаємодія може мати значний вплив на фізичне та психічне здоров'я. Стрес є реакцією організму на будь-який фактор, що вимагає від нього адаптації або реагування. Ця реакція активує автономну нервову систему, яка включає симпатичну та парасимпатичну гілки. Симпатична гілка відповідає за "боротьбу або втечу" і призводить до викиду адреналіну та норадреналіну, що підвищують серцевий ритм.

Під стресом досить часто розуміють стан з сильним нервовим перенапруженням. Як правило, в операторській діяльності людина в такому стані не приступає до виконання своїх функцій, проте такий стан може виникнути безпосередньо в процесі самої діяльності. У загальному представленні про стрес вважають, що він є реакцією організму на сильне відчуття, зумовлене тим чи іншим подразником. Для багатьох випадків стресу мають місце адаптаційні механізми, що нівелюють значні напруження.

Дослідники стресу виділяють такі послідовні стадії його перебігу:

- **Реакція тривоги.** Дія стресора запускає реакцію боротьби, в результаті чого активізується симпатична нервова система, яка мобілізує функціональні резерви для боротьби проти стресу [43]. Типові фізіологічні прояви: прискорене серцебиття і дихання, розширення зіниць, тремтіння рук. В результаті у оператора можуть виникнути труднощі з орієнтуванням.
- **Реакція спротиву (адаптації).** Якщо одразу не вдалося впоратись зі стресом, підвищується артеріальний тиск, зазвичай підвищується стійкість та протидія організму до надзвичайних подразників. Оператор мобілізує волю та бажання подолати виниклу нестандартну ситуацію, нештатні обставини, в нього активізується розумова та м'язова діяльність. Якщо дія стресора на цій стадії припиняється або слабшає, зміни, які він викликав, поступово нормалізуються [44].
- **Реакція відновлення (виснаження).** Якщо організм може протидіяти стресу, тобто його ресурси ще не виснажилися настає процес відновлення. Проте, перебуваючи у фазі виснаження людина вже не має потрібного ресурсу і тому розвивається стійке виснаження усього організму, знову з'являються тривожність. Прояви подібної дезадаптації різноманітні й стосуються емоційної, когнітивної та соматичної сфер [45].

Стрес також пов'язаний із зниженням продуктивності, відсутністю мотивації та неякісним прийняттям рішень. Немає сумнівів у ролі стресу у викликанні вигорання, відстороненості та депресії.

Мікрострес і його вплив. У психологічних публікаціях, що стосуються досліджень стресових явищ, інколи з'являється термін мікрострес. Мікрострес,

як його переважно пояснюють та розуміють, – це миттєва або підсвідома реакція спротиву, що триває дуже короткий час. Її досить складно розуміти. Кожен мікрострес можна розглядати з одного боку, як реакцію на подразник, а з другого – як дію до реакції, що формується на рівні підсвідомості. Початок стресової збудливості називають шоком, оскільки в цей момент розум перестає віддавати точні накази стосовно даної ситуації. Кінцем стресового збудження є заспокоєння.

Тривалість мікростресу від початку і до кінця є дуже короткою – від частки секунди і до кількох секунд. В цей час людина починає миттєво реагувати на те, що відбувається особисто з нею.

Кожен мікрострес різко активізує і мобілізує свідомість на пошук відповідної реакції. Реакція кожної людини на один і той самий подразник може бути різною. Протягом дня мікростреси у вигляді миттєвих реакцій на дрібні подразники накопичуються, а в кінці дня, вони виражаються значною втомою.

Стрес сильно впливає на продуктивність праці, оскільки люди, які перебувають у стресі, приймають неякісні рішення у своїй роботі.

Ці невеликі стреси, хоча самі по собі часто незначні, можуть легко стати нормою, через що їх дуже важко помітити. З часом рівень стресу від них може стати дуже високим. В [46] авторами виділено три типи мікростресів, до яких вони відносять такі:

1. Стреси, які знижують нашу особисту працездатність. Працездатність, як подають автори, – це кількість часу та енергії, яку має людина для керування собою і усіма тими вимогами, з якими стикається на роботі і поза нею. Взагалі кажучи, вони або створюють додаткову роботу, або ускладнюють виконання наявної. Підвищений рівень роботи, особливо відповідальність та психологічні навантаження, також потрапляють до цієї категорії.

2. Стреси, які знижують наші емоційні резерви, як правило, пов'язані з негативними почуттями, такими як занепокоєння, відповідальність, дискомфорт,

3. Стреси, які ставлять під сумнів нашу ідентичність або цінності. Останній набір стресів пов'язаний з проблемами нашої особистості, що

переважно має місце в колективних видах роботи, і переважно пов'язані з несумісністю.

Таким чином, мікростреси хоча і є невід'ємною частиною нашого життя, вони мають суттєвий вплив на результати діяльності людини, особливо при прийнятті високо відповідальних оперативних рішень в складних системах управління, яке здійснюється на основі біжучої інформації, отриманої з екрану монітору.

Одним з таких джерел мікростресу стосовно людини-оператора в інформаційно-пошукових системах може бути різка зміна інформаційного потоку, зумовлена відсутністю негайно потрібної інформації. Таку ситуацію можна пояснити на прикладі операторської діяльності в людино-машинному інтерфейсі пошукової системи. Людина-оператор протягом робочої зміни опрацьовує послідовність зображень сцен реального робочого процесу з метою виявлення потрібних об'єктів для прийняття відповідального рішення.

Протягом деякого часу робота йде в нормальному темпі, потрібні об'єкти легко шукати і виявляти. Оператор перебуває в нормальному психофізіологічному стані, не відчуває дискомфорту і фактично зосереджений на своїй роботі. Але, коли раптом, оператор не бачить шуканого об'єкта – він мобілізує свій внутрішній ресурс лише на пошук цього об'єкта, він знає, що він там є, але не знає де він локалізований.

Оператор усвідомлює що час експозиції зображення сцени з шуканим об'єктом зменшується, його нервово-психологічний стан стає перенапруженим, близьким, можливо, до паніки, він уявляє (бачить) жахливі наслідки, що можуть статися, якщо він не знайде цього об'єкта. Таку ситуацію можна змодельовати включити в надану реципієнту послідовність зображення-тести з об'єктами, які досить важко виявити, що в принципі і має імітувати мікростресові ситуації. Саме в такому плані і має бути організоване експериментальне дослідження.

Адаптивність в галузі інформаційно-пошукових систем вимірювати за допомогою різноманітності завдань. Щоб зрозуміти наскільки оператор може адаптуватися до різних ситуацій використовують різні схеми подання завдань:

- Лише легкі завдання;

- Велика кількість легких завдань чергується з складним;
- Велика кількість важких завдань чергується з одним легким;
- Лише важкі завдання;
- Довільний порядок.

Таким чином ми зможемо перевірити чи змінюється ефективність та успішність роботи оператора в різних ситуаціях та дати оцінку наскільки він адаптивний.

Отже, щоб дати оцінку роботи оператора потрібно мати дані для можливості аналізу його результатів за різних умов, враховуючи вплив фізичного стану самого оператора. Перед початком тестування потрібно заповнити дані про самого оператора: вік, стать, наявність захворювань. Також використати пристрій для зчитування серцевого ритму, зібрати метеорологічні дані про атмосферний тиск (та його зміну).

## 2.2. Вимоги щодо організації експерименту

На даний момент пара «оператор – комп'ютер» є широко розповсюдженою підсистемою практично в усіх автоматизованих інформаційно-пошукових системах, а також в системах управління технічними об'єктами та технологічними процесами. Фактично, ця пара виступає як первинний зв'язок, що несе відповідальність за наслідки прийнятих нею рішень. Основна відповідальність за цю підсистему зазвичай покладається на людину-оператора, чия надійність визначальна для ефективності цієї підсистеми і, отже, всієї системи. Хоча призначення таких підсистем може бути надзвичайно різноманітним, з функціонального погляду у них існує значна кількість спільних аспектів.

*Підсистема пошуку об'єктів.* Формально роботу такої підсистеми, позначимо її як  $L$ , можна подати так. Нехай  $L$  – людина-оператор опрацьовує низку зображень, послідовно надаваних на моніторі, в деякій інформаційно-пошуковій системі. Протягом робочого часу  $L$  переглядає на моніторі послідовність цих зображень стосовно вирішення конкретного завдання, а саме виявлення на кожному з них об'єкта заданого класу. Сукупність зображень  $r_i \in$

$R$ , де  $R = \{r_i : r_1, r_2, \dots, r_n\}$ ,  $i = \overline{1, N}$  – цілочисельні індекси цих зображень, характеризує конкретні сцени реальної робочої ситуації. Виявлені на цих зображеннях об'єкти уваги певного класу визначають подальший розвиток даної ситуації. За виявлення кожного об'єкта оператор  $L$  несе певну відповідальність, знаючи, що не виявлення або несвоєчасне виявлення такого об'єкта веде до вельми негативних наслідків [47].

Нехай  $L$ , тобто людина-оператор, діє на проміжку часу  $[0, T_0]$ , а послідовність цих зображень є єдиним інформаційним джерелом даних для нього, оскільки задає координати  $(x_i, y_i)$  локалізації шуканих на цих зображеннях об'єктів. Іншими словами, оператор  $L$  має виявити об'єкти, локалізація яких на наданих зображеннях визначається координатами  $(x_i, y_i)$  та функція визначається формулою 2.1.

$$Z(t) = O_1(x_1; y_1), O_2(x_2; y_2), \dots, O_N(x_N; y_N). \quad (2.1)$$

Ці об'єкти уваги локалізовані на зображеннях випадковим чином, а відтак, їх виявлення має певну складність  $v_i$ , тобто в кількісному плані послідовність зображень може бути представлена відповідною послідовністю показників складності:  $v_1, v_2, \dots, v_N$ .

У даному дослідженні, саме показники складності відіграють найбільшу роль. Справа в тому, що за низького рівня складності виявлення об'єкта пошуку є досить простим – він як би знаходиться перед оператором, його зображення чітке і не замасковане іншими об'єктами чи текстурою тла. Проте, якщо показник складності великий, то об'єкт пошуку може бути добре замаскованим, зливатися з тлом або сусідніми об'єктами, а тому для його виявлення оператору необхідно мобілізувати увагу, напружити зір і, переважно, за високої відповідальності щодо не виявлення об'єкта, він може перейти в нервово напружений стан, який і спричинює різні рівні стресу.

Час експозиції наданого зображення на інформаційному полі – моніторі, який в першу чергу характеризується засобами відображення реальних ситуацій, тобто зовнішніми пристроями: відеокамерами, спеціальними сенсорами або контролюючою апаратурою може бути різним, але завжди задається експериментом. У залежності від засобів спостереження моменти часу експозиції

зображень цих реальних ситуацій можуть бути регулярними, коли час експозиції та зміни зображень є однаковою тривалістю для всіх зображень або коли ці тривалості є випадковими – подання зображень є нерегулярним.

Локалізація зображення об'єкта пошуку,  $r_i(x_i, y_i), r_i \in R, i = \overline{1, N}$ , яке задане своїми координатами  $(x_i, y_i)$  є зв'язана з моментом експозиції цього зображення  $\tau_i, i = \overline{1, N}$ , причому має місце умова  $t_0 \leq \tau_1 \leq \tau_2 \leq \dots \leq \tau_N \leq T_0$ .

Оскільки потік наданих зображень формується поза людино-машинним інтерфейсом, оператор  $L$  не може змінити ні порядок слідування зображень, ні тривалість їхньої експозиції.

Важливим моментом в системах навчання та професійного відбору кандидатів на операторську діяльність є досягнутий або наявний рівень кваліфікації, який можна формально подати в такий спосіб.

Нехай оператор  $L$  діє на проміжку часу  $[0, T_0]$  і єдиним інформаційним джерелом для нього є надана послідовність зображень. Тоді його кваліфікацію, досвід і знання можна формально подати в такий спосіб.

Нехай кількість  $N' \in N$  переглянутих і опрацьованих зображень в попередніх дослідженнях і навчанні визначається кількістю відрізків часу витрачених на кожне зображення:  $\tau_1 \leq \tau_2 \leq \dots \leq \tau_{N'} \leq 0$ .

Очевидно, що сума індивідуальних значень часу цих відрізків, поділена на їх кількість, характеризує оперативність реципієнта.

Кількість зображень в потоці теперішнього дослідження формально подана формулою 2.2.

$$0 < \tau_{N'+1} \leq \tau_{N'+2} \leq \dots \leq \tau_N \leq T_0 \quad (2.2)$$

Нехай

$$D_L(0) = \{(x_i, y_i), v_i | i = \overline{1, N'}\} \quad (2.3)$$

це усі зображення інформаційного потоку опрацьовані оператором до моменту часу  $t = 0$ . Це означає, що оператор має вже певні знання і деякий досвід про дану діяльність. Можемо також вважати, що  $D_L(0)$ , визначений формулою 2.3., це результат його навчання в сенсі знань про інформаційний потік. Можемо також припустити, що до моменту часу  $t = 0$  оператор  $L$  отримав усю необхідну для

роботи інформацію щодо наданого йому тепер потоку зображень про реальну робочу ситуацію в повному обсязі. Після навчання оператор  $L$  повинен виявити локалізований на наданому зображенні заданий об'єкт за мінімальний проміжок часу.

З другого боку, виявлення об'єкта має певну складність, оскільки локалізація об'єктів на зображеннях, що утворюють цю послідовність, є випадковою. Це означає, що на одних зображеннях об'єкт уваги легко знайти, а на інших – досить складно.

Отже, здійснюючи пошук об'єкта, кожен з яких характеризується певною складністю його виявлення, яка визначена формулою 2.4, оператор витрачає

$$C(x_i, y_i) = C_{v_i}(\tau_i), \quad i = \overline{1, N}, \quad (2.4)$$

відповідний нервово-психічний ресурс. Позначимо цей ресурс  $M_L(t)$ ,  $t \in [0, T_0]$ . Чим більша складність виявлення, тим більше ресурсу витрачено і за високої складності зображень та значної їх кількості оператор може перейти в стресовий стан навіть у випадку одного зображення.

В реальних робочих ситуаціях, як правило, цей ресурс може відновлюватись, наприклад, коротким відпочинком, новою додатковою інформацією, рекомендаціями зі сторони тощо, проте у високо відповідальних системах та за незвичайних умов цей ресурс зразу не відновлюється, тобто потребує тривалого часу.

Отже, імітуючи складність виявлення об'єкта вже на моделях сцен реальних робочих ситуацій можна створювати зображення з відповідним рівнем складності і в такий спосіб імітувати в лабораторних умовах мікростресові ситуації.

Проте варта зробити таке зауваження: стрес проявляється не у показниках часу, а у психофізіологічних показниках. Тут час лише вказує в яких ситуаціях перехід в стресовий стан є найбільш ймовірним.

У експериментальних дослідженнях діяльності людини-оператора в якості вхідної інформації переважно використовують не оригінальні зображення реальних ситуацій, а певним чином оброблені їх копії. Суть такої обробки в тому, що з метою ускладнення пошуку об'єкта уваги на таке зображення накладають



шум, інші об'єкти, які маскують шуканий об'єкт або зображення роблять розмитим або певним чином текстурованим. Крім того, для кількісного оцінювання рівня кваліфікації реципієнта використовують, отримані в попередніх експериментах, дані і встановлюють для кожного такого зображення відповідну шкалу складності пошуку в одиницях часу. Після такої обробки ці зображення вже називають загальноприйнятим терміном тест. В результаті експериментального дослідження з використанням таких зображень-тестів встановлюють індивідуальні кількісні та якісні психологічні показники реципієнта.

Загальною вимогою для будь-якого тесту є визначення особливостей поведінки реципієнта за існуючих умов середовища та поставлених завдань. Тому, орієнтація тестів на конкретні завдання має важливе значення.

Основна складність в оцінюванні кваліфікації людини-оператора за допомогою спеціалізованих зображень-тестів в задачах пошуку і виявлення на них об'єктів заданого класу є те, що одне і те саме зображення різними операторами одного і того ж рівня підготовки, опрацьовується за різний час. Крім того, оператор в різні моменти часу також опрацьовує одне й те саме зображення за різний час.

Суттєве значення при пошуку на цих зображень об'єктів має їх локалізація. Саме в цьому, переважно, полягає складність їхнього виявлення, а також маскувальному впливі з боку інших присутніх об'єктів.

Основною трудностю у розробці зображень-тестів є визначення їх складності, а саме відношення їх доцільності для обробки оператором. Іншими словами, в таких експериментах визначається залежність між «складністю об'єкта - часом виявлення» для кожного оператора. Єдиним показником складності зображення з об'єктом є середній час обробки цього зображення групою операторів-експертів або, принаймні, групою реципієнтів. Таким чином, в системах комп'ютерного тренінгу використовують зображення-тести, які володіють відповідною складністю. Ці зображення можуть бути характеризовані або вже відомими, або додатково визначеними кількісними, статистичними та геометричними характеристиками, включаючи параметри алгоритмів їх створення.

У проведених експериментах зображення-тести структуровані як послідовності сцен, що відтворюють реальний робочий процес з заданою тривалістю експозиції. У таких послідовностях можуть використовуватися ті ж самі зображення, проте вони відрізняються за локалізацією об'єктів на них.

У випадку, коли тривалість експозиції невелика – декілька або більше секунд, реципієнт має можливість виділити та зафіксувати окремі елементи та фрагменти тла. Під час подальших зображень реципієнт витратить час саме на пошук таких елементів, що може значно підірвати об'єктивність отриманих даних.

Отже, з метою відволікання уваги від елементів тла на зображенні-тесті сцени, зображення тла піддають процедурі текстурної стилізації. Термін «текстурна стилізація» вказує на сегментацію зображення тла на фрагменти приблизно однакової кольорової гами, при цьому всі пікселі кожного сегмента замінюються кольором, що представляє середнє значення його кольорів.

У певному відношенні це дозволяє уникнути непотрібного зафіксування особливих елементів фрагментів, оскільки під час пошуку ділянок, однорідних за кольором, вони негайно ігноруються. Текстурна стилізація сприяє більшій однорідності тла у подібних оригінальних зображеннях, в той же час розмиваючи дрібні деталі. Кожне таке зображення-тест ілюструє тло, характерне для імітованої робочої ситуації, фактично виступаючи як одна з сцен робочого процесу.

Отримане таким чином текстуроване зображення тла накладається на зображення шуканого об'єкта, здебільшого змінюючи його положення випадковим чином, з метою ускладнення його виявлення у деяких випадках та полегшення в інших. Розмір об'єктів, як правило, обмежений невеликою площею, переважно в межах 10x10 пікселів одного кольору. Щоб уникнути запам'ятовування фрагментів тла оператором, його зображення стилізується текстурно, що імітує розмитість, низький контраст і загалом однорідність у всіх зображеннях набору. Додатково, завдяки поворотам текстурованого зображення, збільшується різноманітність його подань, оскільки повернуті зображення важко розрізнити між собою. Поворот стилізованих зображень дозволяє розглядати оригінальне та повернуте на 90° чи 180° як два різних, що є важливим при

обмеженому часі. Кількість зображень-тестів з виявленими або невиявленими об'єктами пошуку, у порівнянні з загальною кількістю зображень у даному наборі, служить оцінкою ефективності роботи та, отже, кваліфікації оператора.

У процесі розробки зображень для проведення фахового тестування, спрямованого на оцінку здатності вони виконувати функцію тестів та вимірювати реальний рівень певної властивості чи якості (у даному випадку, ефективності виявлення об'єкта уваги реципієнта), використовується концепція валідності.

Валідність тесту показує, якою мірою він вимірює ту якість (властивість, здатність, характеристику і т.п.), для оцінки якої він призначений.

Виділяють три основних види валідності: змістовну (логічну), емпіричну і концептуальну.

Змістовна валідність означає валідність з точки зору фахівців, тобто тест є валідним, якщо так вважає фахівець. Змістовну валідність слід відрізнити від очевидної, зовнішньої.

Очевидна валідність – це валідність з точки зору дослідника стосовно вимірюваного показника. Це те враження про предмет вимірювання, яке сформувалося в нього при знайомстві з інструкцією і матеріалом тесту. В процесі тестування вона в першу чергу визначає відношення дослідника до даного вимірювання.

Емпірична валідність виявляється у визначенні спроможності тесту служити індикатором чітко визначеної особливості чи форми поведінки особи. Для кількісного вимірювання цієї властивості застосовують обчислення коефіцієнта кореляції між результатами тесту та конкретним зовнішнім критерієм.

Розробка тестів для оцінки якості та ефективності операторської діяльності представляє собою значущу та відповідальну задачу, причому відповідно структуровані тести слугують конкретними метрологічними інструментами. Інформація, що отримана в ході правильно організованої та виконаної процедури тестування за допомогою адекватно розроблених та належно підготовлених тестів, забезпечує високий рівень гарантії об'єктивної оцінки вивчених параметрів операторів інформаційно-пошукових систем.

Очевидно, що подібні зображення-тести повинні відповідати всім критеріям психологічних тестів, оскільки вони залучаються до взаємодії з візуальним аналізатором людини та взаємодіють з його функціональним (психофізіологічним) станом.

Психологічний тест – це стандартизоване, досить часто обмежене у часі випробування, призначене для встановлення кількісних і якісних індивідуально-психологічних особливостей.

Основне завдання тесту – передбачення на його основі майбутньої поведінки досліджуваного.

Зазначені зображення-тести ефективно імітують робочу ситуацію, зосереджуючи увагу реципієнта на вхідній інформації та виключаючи реальне робоче середовище. Вони є психологічними тестами, які вимагають від особи виконання завдань у сценарії, що відрізняється від реальної ситуації. Такий тип тестів застосовується при відборі кандидатів на роботу, зокрема у наборі персоналу операторського персоналу. Крім того, ці зображення-тести мають значущість у процесі атестації персоналу, оцінюючи їх здатність до прийняття адекватних рішень в різноманітних робочих ситуаціях.

На основі розробленої моделі тесту в дисертаційному дослідженні розроблено метод впливу потоку мікрострессорів на діяльність оператора.

### **2.3. Організація експерименту**

Для аналізу параметрів оператора з використанням комп'ютерного тренажера важливим кроком є проведення тестування потенційних операторів.

Під час проведення експерименту потрібно організувати робоче місце, що включає ноутбук або персональний комп'ютер з монітором для надання тестів та маніпуляторами для вводу персональних даних. Також вимогою є вміння виконувати базові завдання за комп'ютером та пройти інструктаж по роботі з розробленим комп'ютерним тренажером.

Для проведення експерименту взято роботу авіадиспетчера. Для симуляції роботи авіадиспетчера використану авіаційну карту. Авіаційна карта – це графічне представлення поверхні Землі, спеціально призначене для потреб

навігаційного використання. Вона включає в себе розмаїття географічної і топографічної інформації, необхідної для забезпечення безпечного та ефективного руху повітряних суден. Повітряні судна в тренажері є об'єктами пошуку. Приклад авіаційної карти зображено на Рис. 2.1.

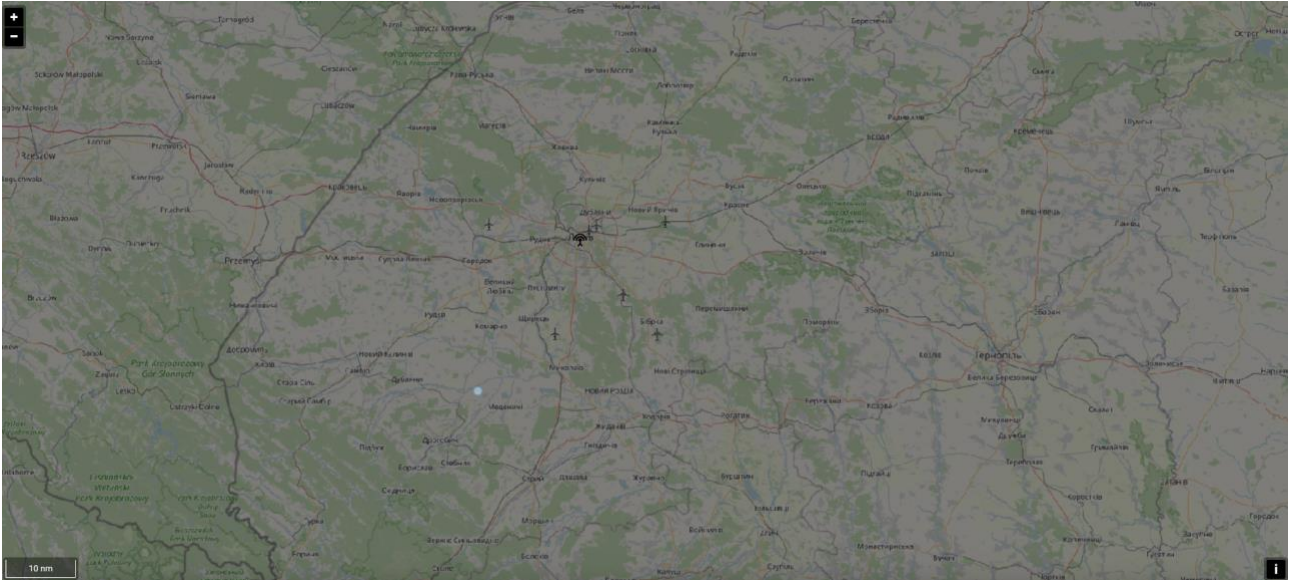


Рис. 2.1. Комп'ютерний тренажер зображення початку роботи

Завданням оператора є виконати пошук заданого об'єкту. З метою наближення виконання завдань до реальних умов на авіаційній карті додано додаткові повітряні судна, а також введено умовні позначення літаків, що підсилюватиме складність виконання завдань. Умовні позначення літаків представляють собою повітряні судна, які мають ідентифікаційний номер літака, до якого авіадиспетчер бажає вислати сигнал, проте на карті мають іншу ідентифікаційну мітку; таким чином, вони фактично виступають в ролі перешкод. Іншим варіантом є ситуація, коли літаки мають аналогічні мітки, але відмінні номери літаків.

Оскільки вихідна позиція користувача має визначальне значення для здійснення процедури тестування, вона повинна залишатися абстрактною з метою виключення можливості запам'ятовування рельєфу карти. Під час кожного етапу тестування оператор може опинитися в різних початкових позиціях [48]. Інформація про початкову позицію доступна лише на етапі початку кожного тесту, спричиняючи уникнення будь-якого запам'ятовування даних про розташування. Важливо відзначити, що вихідна позиція не суттєво відрізняється від позиції об'єкту пошуку, з метою уникнення можливих помилок або

відсутності достатнього часу для виконання процедури пошуку. Загалом користувач негайно опиняється в необхідному секторі, де розташований об'єкт пошуку.

У ході розробки комп'ютерного тренажеру було виконано створення 40 тестів, які належать до одного типу. Всі ці тести виявляють численні загальні риси, об'єднуючи їх в рамках специфічного виду діяльності, яку виконує людина-оператор. Цей аспект виявляється важливим, оскільки головною метою корисного тренажеру є точне відтворення реальної робочої ситуації, причому багато завдань для оператора характеризуються монотонністю.

Протягом процесу тестування кожен потенційний оператор випробовувався в рамках трьох сесій, кожна з яких включала 40 різних тестів. Перша сесія охоплювала послідовне виконання тестів у порядку з 1 по 40, друга - від 40 до 1, тоді як третя передбачала проходження тестів у довільному порядку. Важливість проведення тестування у порядку зростання та спадання складності визначається тим, що кожен окремий тест має свій умовний рівень складності. Такий підхід дозволяє виявити особливості та вплив змін у рівні складності на результати тестування для кожного користувача.

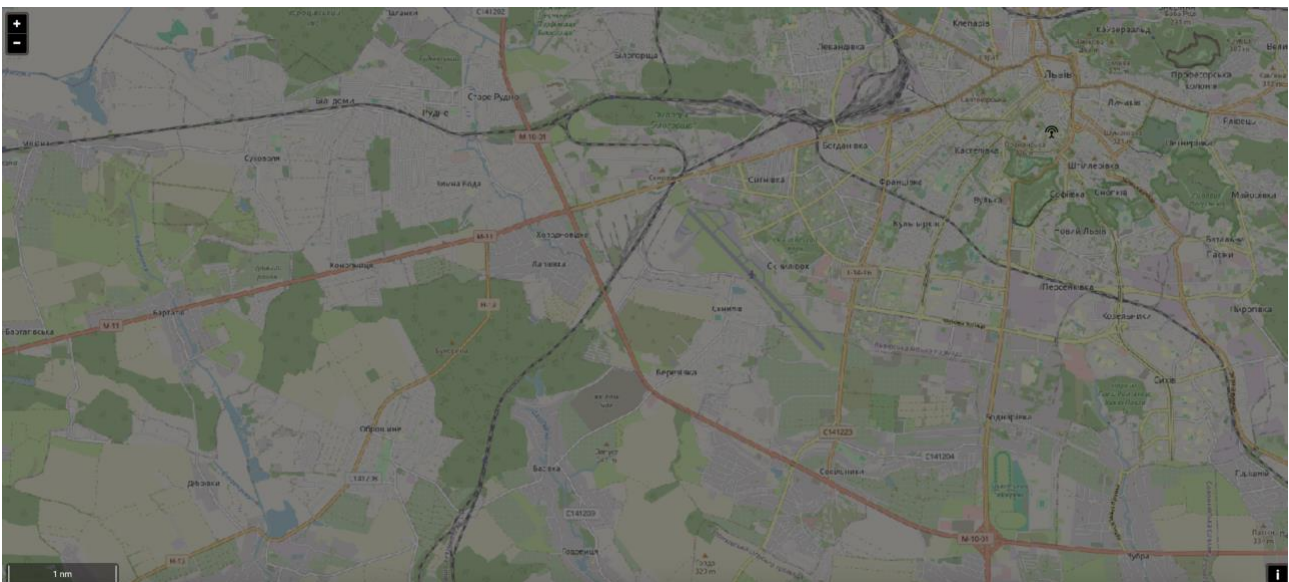


Рис. 2.2. Комп'ютерний тренажер з об'єктом пошуку, що має ефект прозорості

Для оператора новий порядок тестів є практично випадковим при першому проходженні, проте значущість проведення тестування у порядку 1-40 і 40-1 полягає в тому, що кожен тест має свою власну умовну складність, тому

успішність його виконання у порядку змінюється, зростаючи чи спадаючи. Загалом тести розподілені на три категорії умовної складності: тести від 1 до 10 вважаються легкими, тести від 11 до 30 - середніми, а тести від 31 до 40 вважаються складними. Ці категорії відрізняються за кількістю додаткових об'єктів, об'єктів шуму та ступенем прозорості об'єктів пошуку, зображено на Рис. 2.2., які відображаються на карті під час виконання завдань.

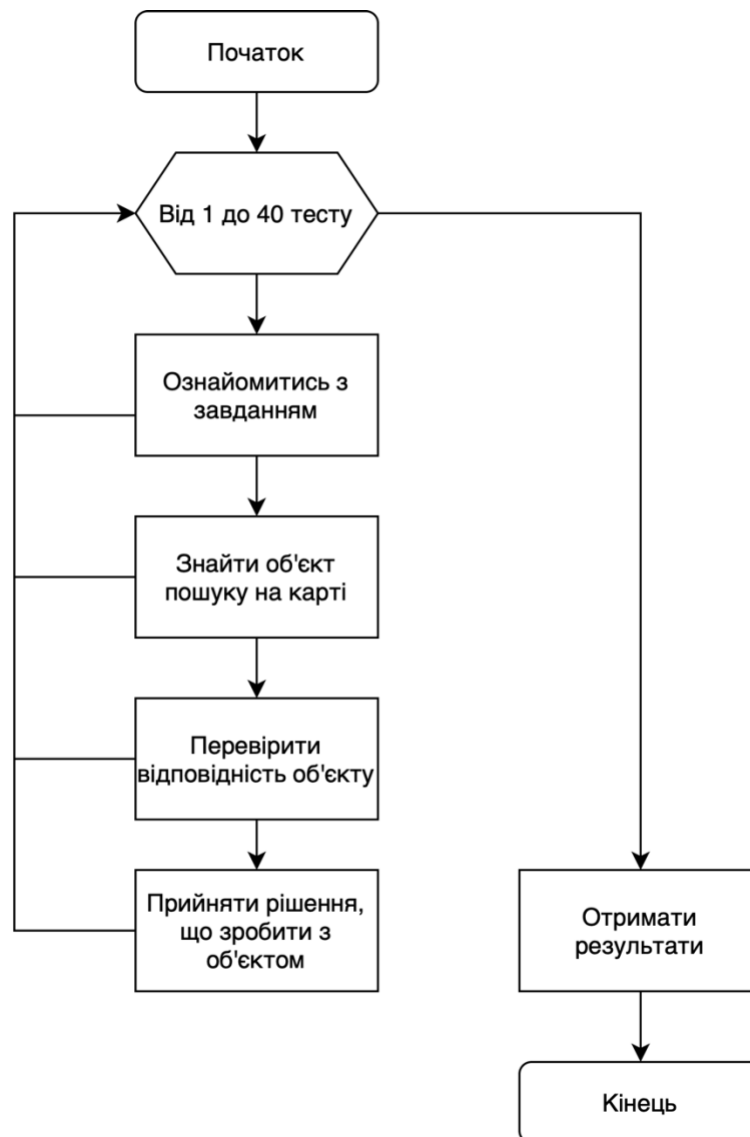


Рис. 2.3. Блок-схема роботи з тренажером

У ході проведення наукових експериментів на авіаційній карті, спрямованих на пошук повітряних суден, здійснюється вимірювання часу виконання кожного тесту з урахуванням обмеження у 30 секунд. Кожен оператор, беручи участь у експерименті, використовував розкмний годинник для вимірювання пульсу під

час виконання завдань на тренажері, при цьому враховувався атмосферний тиск упродовж тривання експерименту. Окрім динамічних параметрів, у тестуванні використовувалися також статичні параметри, такі як вік оператора та його стать. Врахування широкого спектру параметрів дозволяє отримати більш повне та глибоке розуміння впливу різноманітних параметрів на ефективність, точність та час реакції операторів в умовах роботи з авіаційним тренажером.

Блок-схема роботи людини-оператора з тренажером зображена на Рис. 2.3. Під час експерименту людина-оператор виконує чотири підзавдання по яких відстежується час його виконання. Починається експеримент із підзавдання "Ознайомитись з завданням", що становить перший та критичний етап у розумінні завдання для оператора. Наступним етапом є "Знайти об'єкт пошуку на карті", що включає в себе процес активного пошуку вказаного об'єкта. Третє підзавдання полягає в "Перевірити відповідність об'єкту"; під час виконання цього етапу оператор перевіряє правильність вибору об'єкта.

Останнім етапом є "Прийняти рішення, що зробити з об'єктом", зображено на Рис. 2.4. Цей етап передбачає прийняття відповідального рішення стосовно подальших заходів із знайденим об'єктом, таких як передача його відповідній службі або ініціювання сигналу для посадки літака. Загальний процес поділяється на чітко визначені етапи, що сприяє систематизації та ефективності в роботі оператора. У кінці після виконання усіх тестів людина-оператор може отримати результати та ознайомитись з ними.

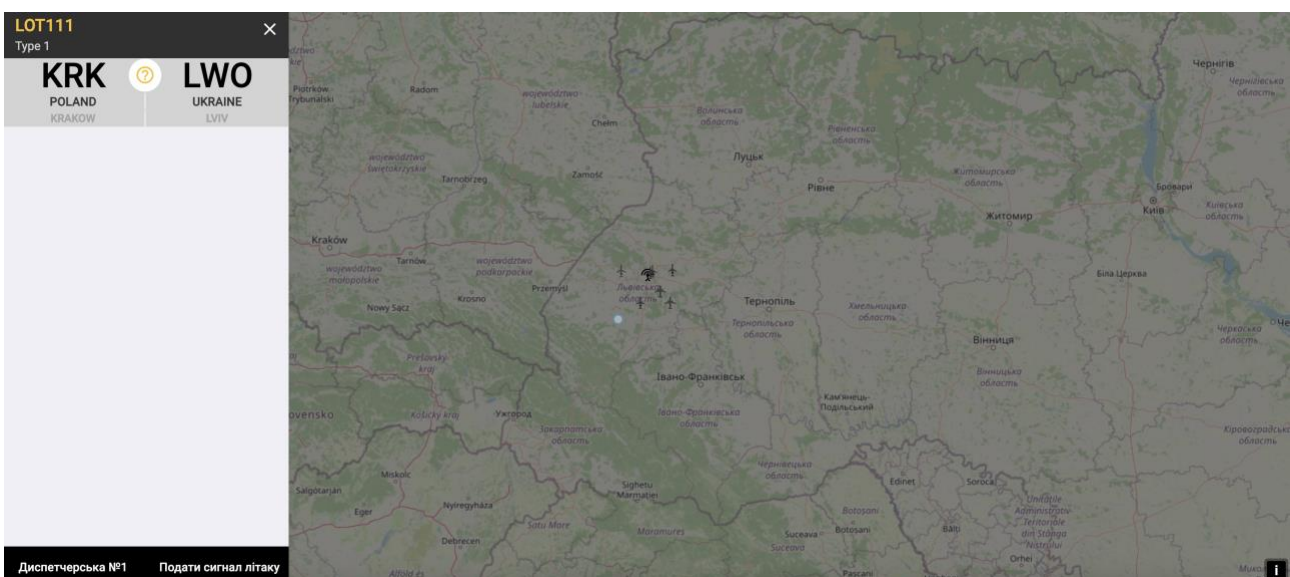


Рис. 2.4. Комп'ютерний тренажер при моменті прийняття рішення



Таким чином побудовано та описано експеримент для роботи людини-оператора за комп'ютерним тренажером. Наведено рисунки, що показують основні сценарії навчання та тестування операторів за тренажером.

#### 2.4. Метод аналізу впливу потоку мікрострессорів на діяльність оператора на основі східцевих функцій

Функція  $U(t - t_0)$  називається узагальненою одиничною функцією. Графік цієї функції, для координати часу  $t = t_0 \neq 0$ , що означає її зсув вправо на  $t_0$ , зображено на Рис. 2.5.

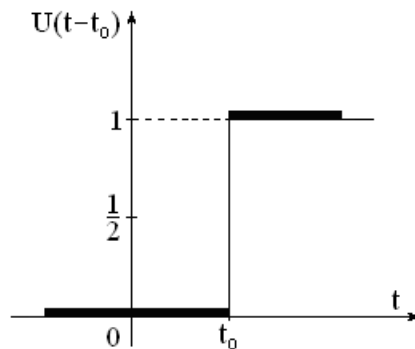


Рис. 2.5. Графіки одиничної функції.

У цьому дослідженні одиничну функцію використано для генерації мікрострессора. З допомогою узагальнених одиничних функцій легко представити локалізований в часі прямокутний імпульс  $f(t)$ , визначений формулою 2.5, який має аналітичний опис у формі обмежень тривалості цього імпульсу

$$f(t) = \begin{cases} a, & t_1 < t < t_2, \\ 0, & t < t_1, t > t_2. \end{cases} \quad (2.5)$$

Функцію  $f(t)$  за допомогою узагальненої одиничної функції подано у формулі 2.6, а графік цієї функції має вигляд зображений на Рис. 2.6.

$$f(t) = a[\eta(t - t_1) - \eta(t - t_2)]. \quad (2.6)$$

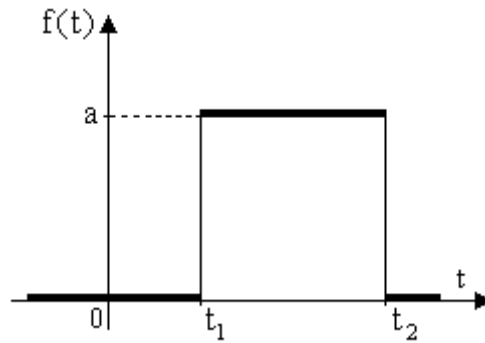


Рис. 2.6 Графік одиничного імпульсу локалізованого в часі.

Східцева функція є математичною моделлю, яка характеризує траєкторію переміщення розгортаючого елемента вздовж осі  $\hat{l}\delta$ , наприклад, вздовж стрічки зліва направо, і вона відображається у вигляді східців, що піднімаються вгору. Ці східці ефективно моделюють мікростреси, які виникають під час проведення тестів. Тривалість кожної східцевої ступені в часі визначається як час проходження одного пікселя, а також час стрибка, що є рівним значенню амплітуди  $a$ . Спільна послідовність таких стрибків, які мають постійну амплітуду і регулярно розташовані в часі, відповідає кусково-лінійній функції, що описує їх траєкторію. У випадку, коли перегляд стрічок відбувається зверху вниз вздовж осі  $Oy$ , тобто стрічка за стрічкою апертура опускається, таку траєкторію можна описати подібною, але спадною функцією, проте відстань між сусідніми стрибками в часі тепер визначається тривалістю всіх пікселів, які складають стрічку. Такі функції, що репрезентують східцеподібні траєкторії, отримали назву східцевих функцій і є вираженими у вигляді одиничних узагальнених функцій. Для зростаючої зліва направо східцевої функції  $f_{\zeta\delta}(t)$  з  $n$  східців і амплітудою кожної східця  $a$ , зображеної на Рис. 2.3, аналітичне представлення є формула 2.7.

$$f_{\zeta\delta}(t) = \begin{cases} 0, & t < 0 \\ n a, & t_n < t < t_{n+1}. \end{cases} \quad (2.7)$$

де  $n = 0, 1, 2, \dots$

Зростаючу східцеву функцію можна за допомогою узагальненої функції представити сумою наведеною у формулі 2.8.

$$f_{\zeta\delta}(t) = a[\eta(t - t_0) - \eta(t - t_1) + 2\eta(t - t_1) - 2\eta(t - t_2) + 3\eta(t - t_2) + \dots + (n - 1)\eta(t - t_{n-2}) - (n - 1)\eta(t - t_{n-1}) + n\eta(t - t_{n-1}) - n\eta(t - t_n)]. \quad (2.8)$$

Групування доданків дає таку модель послідовності експозиції тестів та часу їх пошуку:

$$f_{\zeta\delta}(t) = a \sum_{i=0}^{n-1} \eta(t - t_i) - n\eta(t - t_n) \quad (2.9)$$

де  $a = 1$  визначає нормоване значення амплітуди сідців, а  $i \in Z^+$ .

Графік цієї функції зображено на Рис. 2.7. Другий доданок в формулі 2.9 при рівності аргументів  $t_i = t_n$  обнуляє значення суми, тобто, значення суми існує і є пропорційне індикативній змінній  $i$  до того часу, поки  $0 \leq i < n$ , а при  $i \geq n$ ,  $h(t) = 0$ . У цей спосіб строго обмежується кількість сідців до  $n$ .

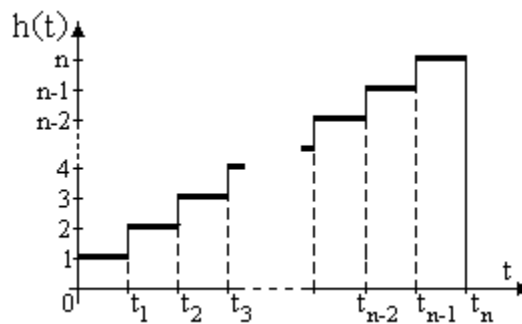


Рис. 2.7. Зростаюча сідцева функція

Маючи дані про зміну стану в роботі оператора та часові інтервали можна побудувати одиничну функцію для зміни станів людини-оператора комп'ютерного тренажеру. Побудувавши графік для багатьох станів оператора та повторень виконання тестів отримаємо періодичну сідцеву функцію. Це дає змогу побачити зміни та залежності між станами оператора та різними тестами або повторами того ж тесту кілька разів.

Побудуємо періодичну сідцеву функцію для перших трьох тестів під час роботи з тренажером, зображено на Рис. 2.8., а також накладемо обернену сідцеву функцію зі зміщенням на один крок. Кожен крок – це стан в якому перебуває оператор. Останнім кроком в кожному тесті є прийняття рішення, саме в цей момент найбільше стресу, так як від одного кліку залежить весь тест. Побудувавши сідцеву і обернену функцію виділяємо різницю для того, щоб побачити різницю і отримати моменти, у яких оператор перебуває перед

прийняттям рішення. На Рис 2.9 зображено зрізи часу моменту прийняття рішення, саме вони відображають стресові ситуації. Коли імпульси великі то люди більше часу справляється зі стресом.

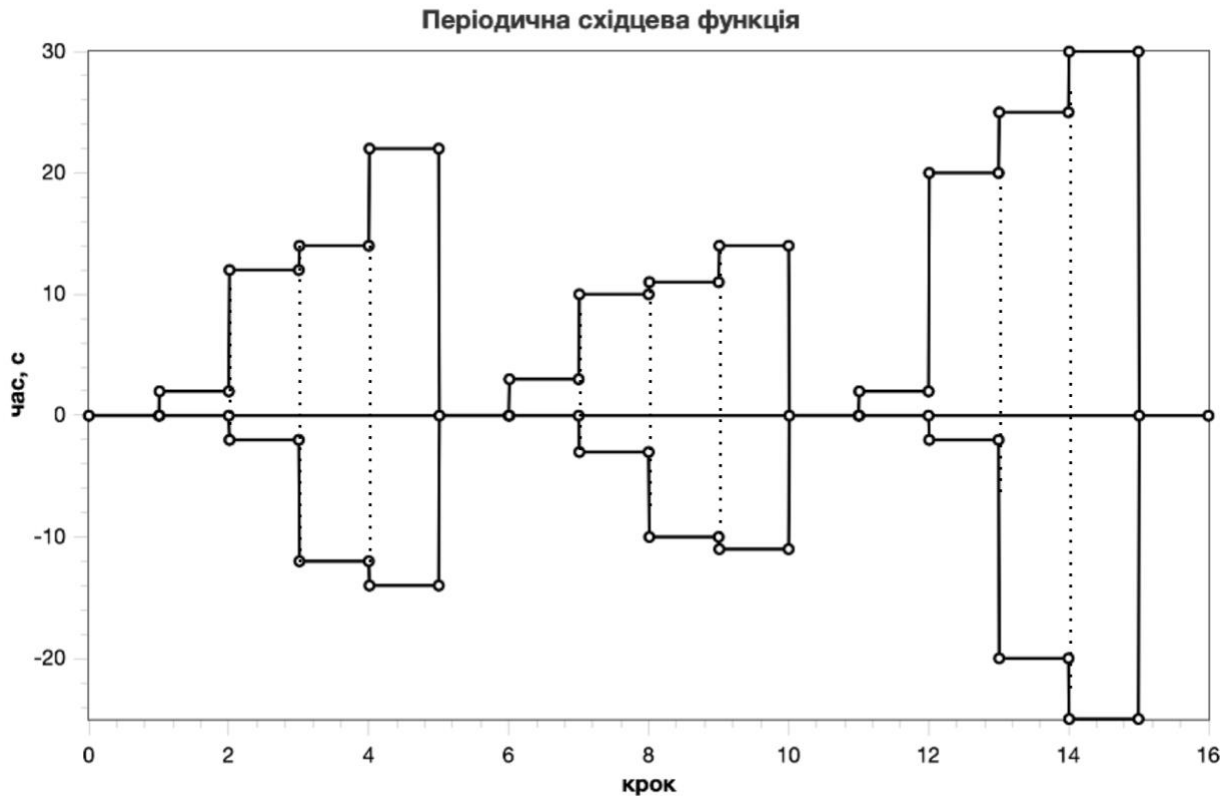


Рис. 2.8. Періодична східцева функція.

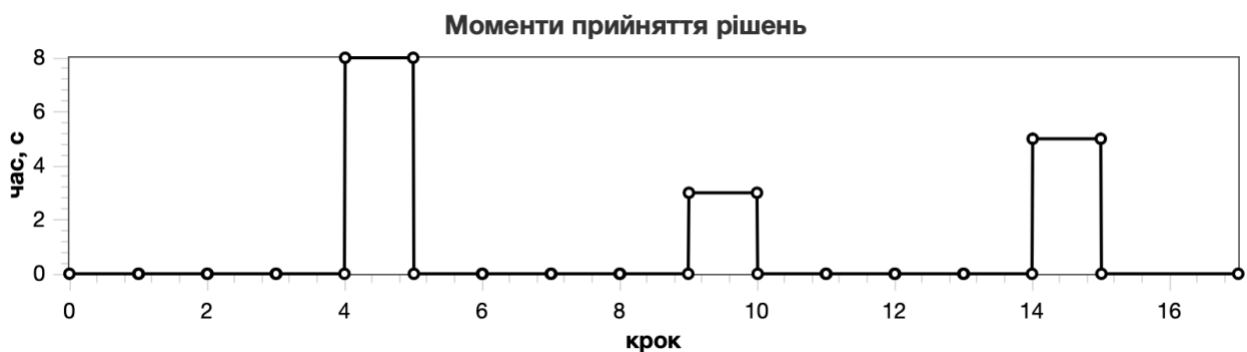


Рис. 2.9. Зрізи моментів прийняття рішень.

Наступним кроком побудуємо східцеву функцію проходження п'яти тестів. Кожен тест має 30 секунд на виконання, що відображає оранжева лінія на Рис. 2.10., а синя це реальний час проходження тесту. По осі x йде загальний час роботи з тренажером, а по осі y час проходження одного тесту. З цього графіку

можна виокремити час відпочинку та побудувати одиничні імпульси, коли людина-оператор перебуває у стані спокою. На Рис. 2.11. зображено імпульси відпочинку. На третьому тесті час відпочинку був 15 секунд, що показує великий імпульс, після чого імпульс став менший, що може говорити, що людина розслабилась або просто попав важчий тест. Загалом оптимальний випадок, коли розмір імпульсів є приблизно однаковий для всіх тестів, бо це дає можливість прогнозувати роботу людини та мати рівномірне навантаження.

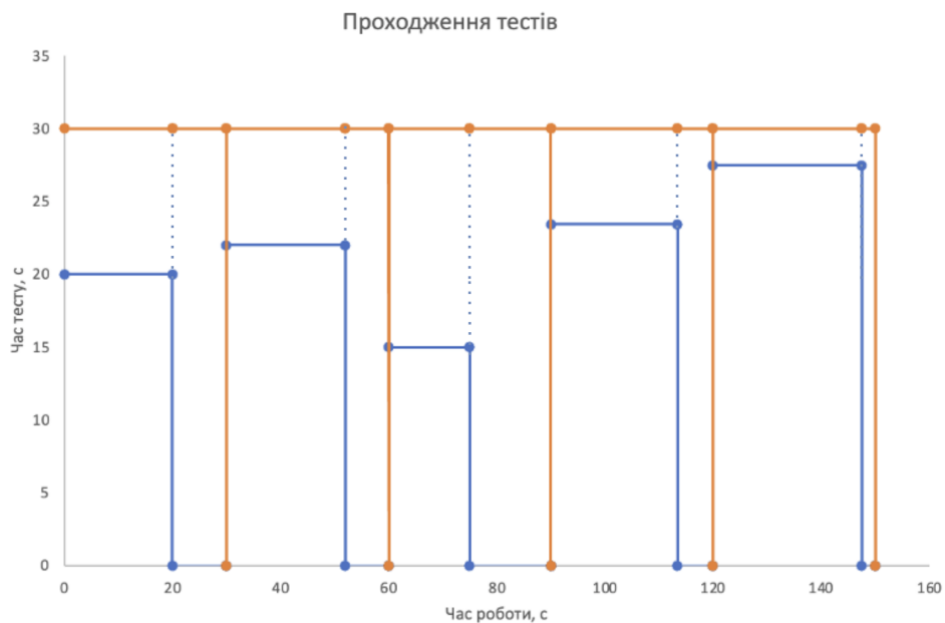


Рис. 2.10. Східцева функція проходження тесту.

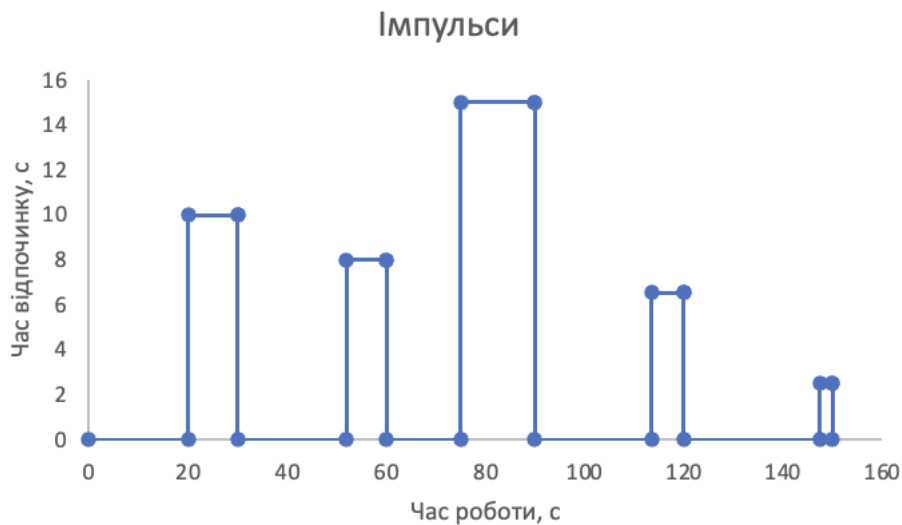


Рис. 2.11. Імпульси відпочинку.

Побудовані східцеві функції по етапам роботи людини-оператора за комп'ютерним тренажером. На виділених імпульсах виокремлено моменти прийняття рішень, що відповідають за стресовий час, а також імпульси відпочинку, що вказують коли відбувалась та скільки тривала зміна уваги. Ці вказують на стресостійкість оператора та здатність працювати прогнозовано. Залежність коливань від коливань висоти імпульсу вказує наскільки стабільно та прогнозовано оператор виконує свою роботу.

### **Висновки до другого розділу**

Було вивчено та проаналізовано параметри людини-оператора комп'ютерного тренажеру та їх вплив на роботу, точність, ефективність. Вибрано параметри для дослідження їхнього впливу на результат.

Проаналізовано математичну модель та поняття тесту для комп'ютерного тренажеру. Виділено етапи пошуку об'єкта, етапи складності його ідентифікації, етап обробки об'єкту пошуку. У даному розділі розроблено комп'ютерний тренажер для аналізу параметрів оператора. Наведено основні особливості тренажеру, його можливості та експеримент, що був проведений.

Розглянуто східцеву функцію як процес роботи з тестом. Виділено моменти часу  $t$  для побудови функції та розгортки періодичної функції. Побудовано східцеву функцію та зміщену обернену для виділення моментів стресу. Побудовано імпульси відпочинку оператора при роботі з тренажером.

## РОЗДІЛ 3. РОЗРОБКА МЕТОДІВ МОДЕЛЮВАННЯ ОПЕРАТОРСЬКОЇ ДІЯЛЬНОСТІ

У розділі приведено методи, використані для обробки даних та їх основні характеристики. Розроблено метод-ансамбль для підвищення точності класифікації. Проведено пошук аномалій серед параметрів оператора, що впливають на результати роботи.

### 3.1. Метод аналізу персональних даних оператора та результатів проходження тестів

Методи аналізу даних – це процес виявлення залежностей, обробки, та використання даних для прийняття обґрунтованих рішень. Вони включають збір, очищення, візуалізацію, статистичний аналіз та моделювання, і використовуються у багатьох галузях для вивчення та прогнозування [49].

Попередня обробка даних перед використанням методів аналізу полягає у зборі, очищенні та підготовці даних [50]. Це включає видалення відсутніх значень, обробку аномалій, нормалізацію та перетворення даних для підготовки до подальшого аналізу та моделювання.

Під час аналізу даних використовується набір даних, зібраний під час виконання тестів на комп'ютерному тренажері. Важливим кроком перед аналізом даних є їх візуалізація, що допомагає краще зрозуміти особливості розподілу даних та де саме можуть бути аномалії чи неточності. Для візуалізації розподілу даних використаємо гістограму. У датасеті немає пропусків чи некоректних даних, ці дані були зібрані спеціально для подальшого дослідження. Проте аномалії можуть бути присутні, так як всі ці дані отримані з діяльності людей-операторів. Загальна кількість даних 6000 записів. При аналізі використовуються 6 властивостей:

- testOrder (порядок тесту) – від 1 до 40,
- responseTime (загальний час відповіді на тест) – має мінімальне значення 5.01 і максимальне 30.0, описова статистика подана в Таблиці 3.1. та доповнена у додатках,

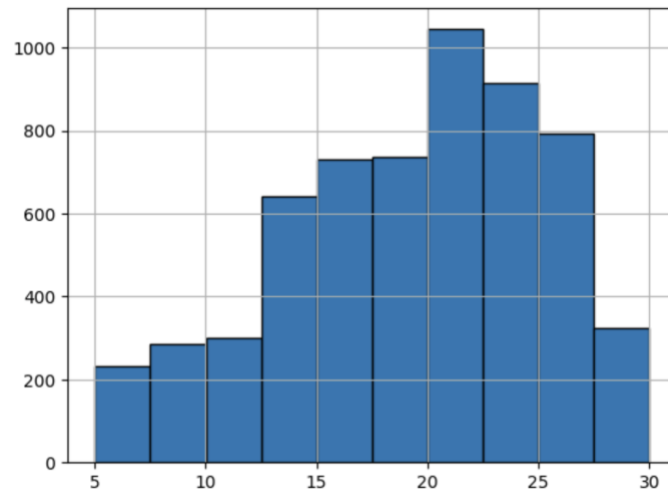


Рис. 3.1. Гістограма розподілу часу роботи з тестом

- `heartrate` (пульс) – має значення від 51 до 173,

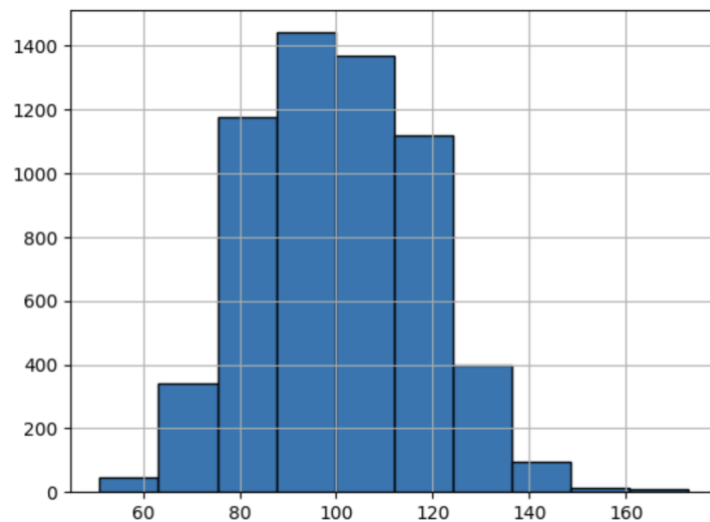


Рис. 3.2. Гістограма розподілу пульсу

- `complexity` (складність) – складність від 1 до 3,
- `atmosphericPressure` (атмосферний тиск) – розподілився на проміжку від 752 до 757,
- `old`(вік) – набуває значень від 23 до 63,
- правильність виконання тесту відображає `success` (має значення `yes` або `no`).

Якщо дивитися за порядком тесту то кожен тест людина-оператор проходила 3 рази. Складність тесту поділена на 3 рівні 1-го рівня складності є 10 тестів, 2-го рівня – 20, 3-го – 10. Успішність виконання тесту розділена наступним чином:



успішне виконання (yes) – 4344, помилка (no) – 1656. У таблиці 3.1 описано часові результати тестів та доповнено для усіх операторів у додатку Б.1.

Таблиця 3.1. Часові результати проходження тестів

Номер оператора	Мін, с	Q1, с	Медіана, с	Q3, с	Макс, с
1	5.49	15.6	21.03	24.51	29.79
2	5.77	16.16	21.47	25.01	29.74
3	6.26	16.81	20.69	23.24	29.17
...	...	...	...	...	...
50	5.04	15.02	19.5	24.53	29.35

Схема аналізу даних передбачає дві стадії:

1. Попередній аналіз даних,
2. Аналіз даних.

Попередній аналіз даних здійснено шляхом застосування моделей машинного навчання для відбору важливих ознак. Аналіз даних реалізовано на основі ансамблювання моделей машинного навчання, використано в підрозділі 3.4.

### 3.1.1. Логістична регресія

Логістична регресія – це статистичний метод машинного навчання, який використовується для моделювання й передбачення ймовірностей бінарних або багатокласових подій [51]. Вона знайшла широке застосування в різних галузях, включаючи медицину, біологію, економіку, маркетинг, соціологію та багато інших. Логістична регресія – це метод, який допомагає вирішити завдання класифікації та оцінки ймовірностей.

Основні аспекти логістичної регресії:

- *Завдання.* Логістична регресія використовується для вирішення задачі бінарної або багатокласової класифікації. У випадку бінарної класифікації, ми маємо два класи, і ми спробуємо визначити, до якого класу належить кожен приклад даних. У випадку багатокласової класифікації, ми маємо більше ніж два класи, і ми повинні призначити кожному прикладу один з цих класів.

- *Функція логістичної регресії.* Основною частиною логістичної регресії є логістичка функція представлена формулою 3.1, яка перетворює лінійну комбінацію вхідних ознак в імовірність.

$$P(Y = 1|X) = \frac{1}{1+e^{-(\beta_0+\beta_1X_1+\beta_2X_2+\dots+\beta_pX_p)}}, \quad (3.1)$$

де  $P(Y = 1|X)$  – імовірність того, що подія  $Y$  належить до класу 1 при заданих вхідних значеннях  $X$ ,  $e$  – базове число (приблизно 2.71828),  $\beta_0, \beta_1, \beta_2, \dots, \beta_p$  – параметри моделі, які оптимізують під час навчання,  $X_1, X_2, \dots, X_p$  – вхідні ознаки [52].

- *Оптимізація параметрів.* Параметри моделі  $\beta_0, \beta_1, \beta_2, \dots, \beta_p$  оптимізуються під час навчання. Зазвичай це робиться за допомогою методу максимальної правдоподібності або інших оптимізаційних методів, таких як градієнтний спуск. Мета полягає в тому, щоб знайти значення параметрів, які мінімізують помилку класифікації на тренувальному наборі даних.
- *Функція втрат.* Функція втрат визначає, наскільки добре модель прогнозує класи. У випадку логістичної регресії, типовою функцією втрат є логарифмічна функція втрат [53].
- *Навчання моделі.* Під час навчання моделі знаходяться такі параметри  $\beta_0, \beta_1, \beta_2, \dots, \beta_p$ , які мінімізують функцію втрат. Це може бути зроблено за допомогою різних методів оптимізації, таких як градієнтний спуск або алгоритми другого порядку.
- *Валідація та оцінка моделі.* Після навчання моделі важливо оцінити її ефективність. Це може бути зроблено за допомогою різних метрик, таких як точність, відзив, точність та F1-міра. Модель також може бути перевірена на тестовому наборі даних, щоб переконатися, що вона генералізується на нові дані.
- *Робота з вхідними ознаками.* Важливо вибирати та інженерувати вхідні ознаки, які мають найбільший вплив на якість моделі. Це може включати в себе відбір ознак, стандартизацію та нормалізацію даних, а також створення нових ознак з існуючих.

- *Використання у реальних задачах.* Логістична регресія добре підходить для задач, де необхідно передбачити ймовірності класифікації, такі як прогнозування ймовірності покупки товару, ймовірності хвороби, або визначення ймовірності погоди [54].

Для задачі класифікації з використанням логістичної регресії використали 10-тикратну кросвалідацію. У результаті роботи логістичної регресії отримали точність 84%, що є гарним показником для такого методу.

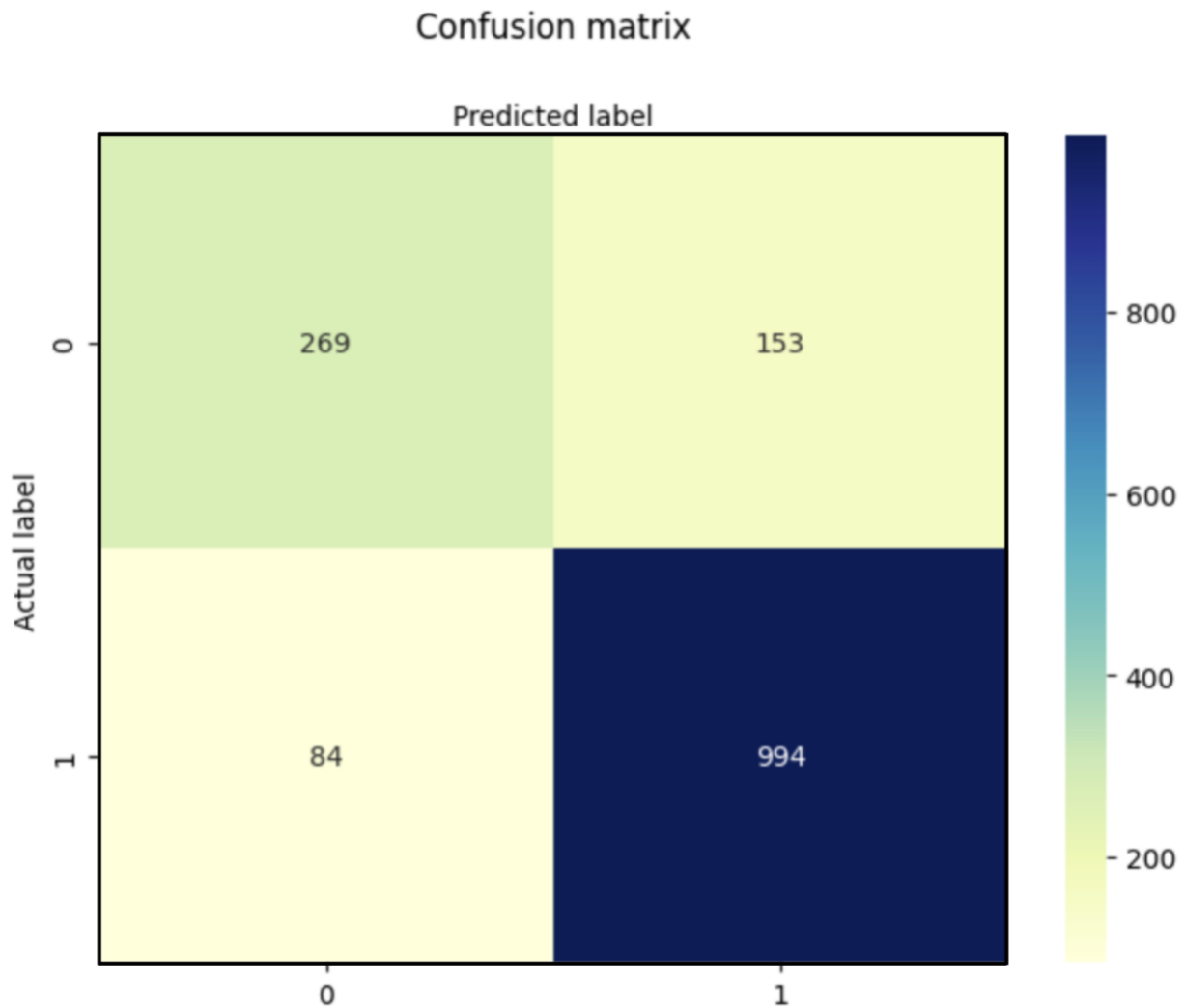


Рис. 3.3. Матриця плутанини

Матриця плутанини – це інструмент прогнозувальної аналітики. Зокрема, це таблиця, яка відображає та порівнює фактичні значення з прогнозованими значеннями моделі. Розглянемо детальніше матрицю плутанини на Рис. 3.3. (confusion matrix). Результати визначення для відповідей yes маємо 994

правильно класифіковано і 84 – ні, для відповідей по 269 правильно і 153 – ні. Це дає нам точність 89% для yes та 69% для no.

### 3.1.2. Метод xgboost

XGBoost (eXtreme Gradient Boosting) – це потужний метод та бібліотека машинного навчання, яка використовується для задач класифікації та регресії [55]. Вона широко використовується у багатьох галузях, включаючи фінанси, біологію, аналіз даних, рекомендаційні системи та багато інших. XGBoost став одним із найпопулярніших і ефективних алгоритмів машинного навчання завдяки своїй високій точності та швидкості навчання [56].

Докладніше розглянемо основні аспекти XGBoost:

- *Гرادієнтний бустінг.* XGBoost використовує техніку градієнтного бустінгу для підвищення якості прогнозів моделі. Градієнтний бустінг – це ансамбльний метод, який об'єднує результати багатьох слабких моделей, таких як дерева рішень, для покращення точності передбачень.
- *Використання дерев рішень.* Основою XGBoost є дерева рішень. Кожне дерево рішень - це слабка модель, яка працює з підмножиною вхідних даних. Деревя об'єднуються, і результати їх роботи враховуються в остаточних прогнозах.
- *Робота з різними видами даних.* XGBoost може використовуватися для різних видів даних, включаючи числові, категоріальні та текстові ознаки. Він має вбудовані методи обробки та кодування цих видів ознак для покращення якості моделі.
- *Важливість ознак.* XGBoost може обчислити важливість кожної ознаки у моделі. Це дає можливість визначити, які ознаки мають найбільший вплив на прогнози.
- *Параметри моделі.* XGBoost має багато параметрів, які можна налаштувати для покращення якості моделі та управління її поведінкою. Деякі з них включають глибину дерев, швидкість навчання, кількість дерев, регуляризацію та багато інших.

- *Крос-валідація.* Для оцінки якості моделі можна використовувати крос-валідацію. Цей метод дозволяє оцінити точність моделі на різних наборах даних та підвищити її стійкість до перенавчання [57].
- *Використання в реальних задачах.* XGBoost застосовується в багатьох різних галузях та завданнях, включаючи фінанси, медицину, рекомендаційні системи, обробку природних мов, аналіз зображень та багато інших. Він широко використовується для прогнозування, класифікації, регресії та інших завдань.
- *Розширення та підтримка.* XGBoost має різні розширення та інтеграції, які роблять його досить гнучким. Він підтримується спільнотою та активно розвивається.

Узагальнюючи, XGBoost – це потужний алгоритм машинного навчання, який володіє високою точністю та швидкістю навчання. Він використовує градієнтний бустінг з деревами рішень як основу, що робить його відмінним вибором для багатьох завдань аналізу даних та прогнозування [58].

Перед навчанням з XGB класифікатором використаємо алгоритм пошуку гіперпараметрів для моделі Grid Search. Задамо пошук для наступних параметрів: `learning_rate`, `min_child_weight`, `colsample_bytree`. Такий пошук зможе покращити точність для класифікації. Маємо точність навчання 0.891 при використанні параметрів за замовчуванням, а при застосуванні параметрів `learning_rate=0.1`, `colsample_bytree=1`, `min_child_weight=3` F1 score становить 0.9.

### 3.1.3. Метод дерево рішень

Метод побудови дерева рішень (`DecisionTreeClassifier`) – це один з популярних алгоритмів машинного навчання, який використовується для завдань класифікації [59]. Він базується на побудові дерева рішень, яке допомагає приймати рішення на основі вхідних ознак і навчальних даних [60]. Дерево рішень є графічною моделлю, де кожен вузол представляє рішення на основі певної ознаки, а листя дерева відповідає класифікаційному результату. Основні характеристики та принципи роботи `DecisionTreeClassifier`:

1) *Побудова дерева рішень:*

Початково вибирається корінь дерева, який відповідає всій навчальній вибірці. Дерево рішень будується шляхом рекурсивного розбиття навчальної вибірки на менші підвибірki на кожному внутрішньому вузлі. Вузли дерева обираються так, щоб максимізувати інформативність поділу, наприклад, за допомогою критерію Джині або ентропії.

2) *Критерії розбиття.* Для вибору оптимального розбиття на кожному вузлі використовуються критерії якості, такі як критерій Джині, ентропія або помилка класифікації. Критерії оцінюють, наскільки добре поділ розподілений по класах в підвибірках.

3) *Рекурсивна побудова.* Дерево будується рекурсивно, доки не виконується одна з зупинок:

- Дерево досягає максимальної глибини;
- Вузол містить менше за мінімальну кількість прикладів;
- Вузол містить всі приклади одного класу.

У кожному вузлі обирається оптимальне розбиття.

4) *Важливість ознак.* Класифікатор обчислює важливість кожної ознаки на основі того, як часто ця ознака використовується для розділення вузлів і як великі зміни в якості моделі вона вносить. Важливість ознаки використовується для ранжування ознак за їх внеском у класифікацію.

5) *Вирішення проблеми перенавчання.* Дерева рішень можуть легко перенавчатися, тобто робити дуже складні рішення, які не генералізуються до нових даних. Щоб запобігти перенавчанню, можна використовувати параметри, такі як максимальна глибина дерева, мінімальна кількість прикладів у вузлі, мінімальна кількість прикладів для розгляду поділу тощо.

6) *Відомості про дерево.* Дерево може бути відображене для інтерпретації та аналізу прийнятих рішень. Можна вивести структуру дерева, включаючи вузли та рішення, прийняті на кожному рівні.

7) *Застосування.* DecisionTreeClassifier застосовується в багатьох галузях, включаючи медицину, фінанси, маркетинг, біологію та інші. Він підходить для завдань класифікації та може бути використаний як самостійний класифікатор або в ансамблі з іншими моделями [61].

8) *Обмеження*. Дерева рішень схильні до перенавчання на шумні дані, і можуть бути не такими точними як інші моделі, такі як випадковий ліс або градієнтний бустінг.

9) *Гіперпараметри*. `DecisionTreeClassifier` має ряд гіперпараметрів, таких як критерій розбиття, максимальна глибина дерева, мінімальна кількість прикладів в вузлі та інші, які можуть бути налаштовані для оптимізації моделі.

У підсумку, `DecisionTreeClassifier` – це потужний інструмент для завдань класифікації, який використовує дерева рішень для прийняття рішень на основі вхідних ознак та навчальних даних. Він підходить для різних застосувань та може бути вдосконалений налаштуваннями гіперпараметрів, щоб досягти оптимальної точності та генералізації.

Для методу класифікації побудови дерева рішень розділено, що тестові дані мають 30%, а дані на яких класифікатор вчиться розпізнавати дані становлять 70%. Дані розбиті у довільному порядку, щоб відкинути залежність між результатами послідовних тестів.

Для кращого розуміння, які саме властивості та як впливають на результати навчання маємо гістограму. Пульс має значення 75.4%, вік – 8.5%, порядок тесту – 5.8% та складність – 10.3%. Зображено на Рис. 3.4.

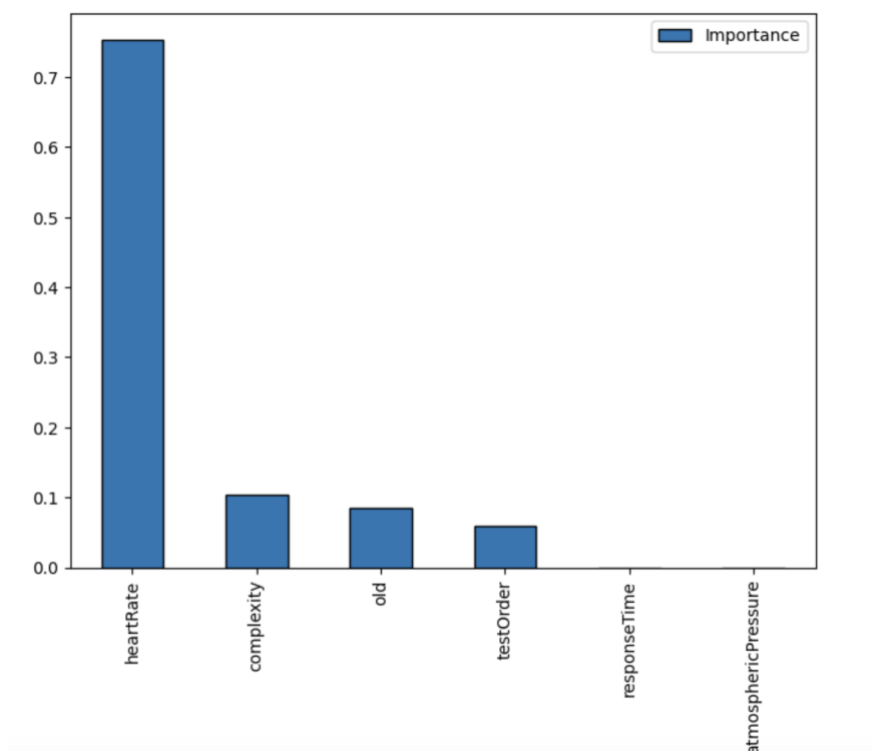


Рис. 3.4. Розподіл важливості властивостей

У результаті навчання отримує точність для тестових даних 85%, а для усіх даних 85% та отримуємо результат F1 score 85%. Це досить добрий результат, а особливо як відносно не великого набору даних. Для класифікатора використано оптимальні параметри  $\text{max\_leaf\_nodes} = 10$  та  $\text{max\_depth} = 10$ . При інших варіантах використання параметрів точність була нижча.

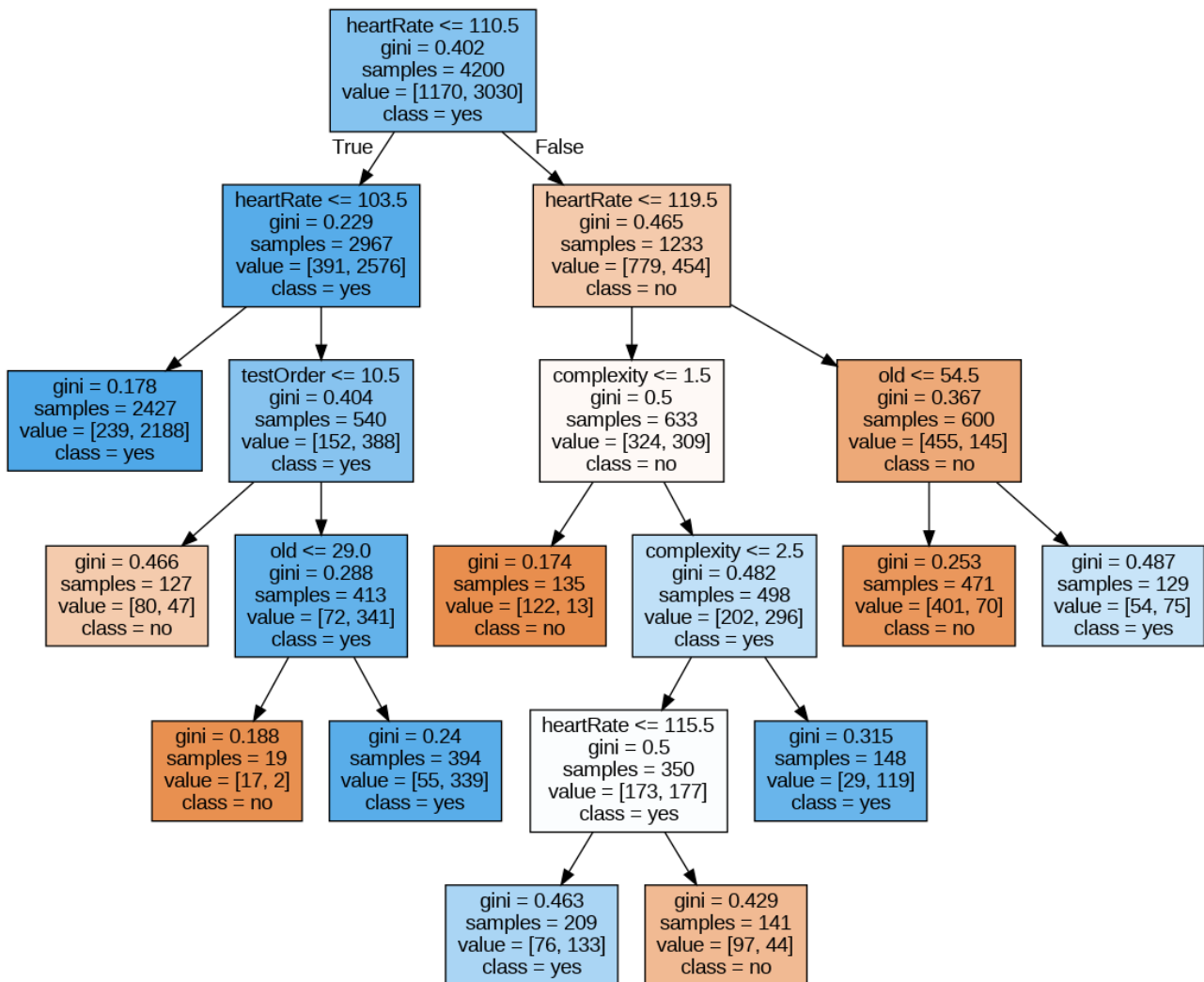


Рис. 3.5. Дерево рішень

На Рис. 3.5. побудовано дерево рішень для моделі, на якому можна побачити як саме визначається чи буде правильна відповідь чи ні. На цьому дереві видно, що нема випадків де до вузла потрапили дані, що повністю підпадають під одне правило, коефіцієнт gini(коефіцієнт розподілу) має найменше значення 0.174. Більшість листків мають значення gini більше 0.4, у певних вузлах значення дорівнює 0.5, що є найкращим значення для розподілу у вузлах.



## 3.2. Кластеризація

Методи кластеризації даних – це група алгоритмів в машинному навчанні та аналізі даних, які призначені для розділення схожих об'єктів у наборі даних на групи або кластери [62]. Кластерний аналіз вирішує задачі: кластеризації – розділення об'єктів на групи на основі подібності між ними; сегментації ринку – допомагає розділити ринок по групах та пріоритетах, що допомагає зрозуміти потреби різних груп людей; виявлення аномалій – дає можливість знайти аномалії чи викиди з загальної картини; та інші задачі. Методи кластеризації даних можуть бути використані в різних галузях, включаючи бізнес, науку, медицину, комп'ютерну графіку та багато інших, і вони допомагають виявити закономірності, групи або змінити в наборах даних для прийняття обґрунтованих рішень [63].

### 3.2.1. Метод к-середніх

Метод к-середніх (K-means) – це один з найпопулярніших алгоритмів кластеризації в машинному навчанні та аналізі даних [64]. Він призначений для групування схожих об'єктів у кластери на основі їхнього схожості. Алгоритм K-means є досить простим та швидким у реалізації, і він знаходить застосування в різних галузях, таких як маркетинговий аналіз, біологія, комп'ютерний зір тощо. Він вимагає вказання кількості кластерів (K), у які дані будуть розділені.

Основні кроки алгоритму K-means:

- 1) *Ініціалізація центрів кластерів.* Починаючи, алгоритм обирає початкові центри кластерів випадковим чином або на підставі якихось вихідних припущень.
- 2) *Призначення до кластерів.* Для кожного об'єкта даних обчислюється відстань до всіх центрів кластерів, і об'єкт призначається до кластера з найближчим центром.
- 3) *Перерахунок центрів кластерів.* Після призначення всіх об'єктів до кластерів обчислюються нові центри для кожного кластера, як середнє значення всіх об'єктів у кластері.

- 4) *Повторення.* Кроки 2 і 3 повторюються до тих пір, поки об'єкти не припинять змінювати своє приналежність до кластерів або досягнута певна зупинка.
- 5) *Завершення і вивід результатів.* Коли алгоритм завершує свою роботу, кожен об'єкт має приналежність до певного кластера, і можна вивести результати кластеризації [65].

Переваги методу K-means:

- Простий у реалізації та розумінні.
- Відмінно підходить для великих наборів даних.
- Швидкий та ефективний.

Недоліки методу K-means:

- Вимагає попереднього визначення кількості кластерів (K).
- Чутливий до початкових умов при ініціалізації центрів.
- Може давати різні результати при різних запусках (через випадкову початкову ініціалізацію).
- Не підходить для деяких типів даних зі складними формами кластерів або великою розмірністю.

Проведемо кластерний аналіз за методом K-means. Та бачимо як дані розподілились між двома кластерами. Кластери розділились за принципом успішного і неуспішного виконання тесту.

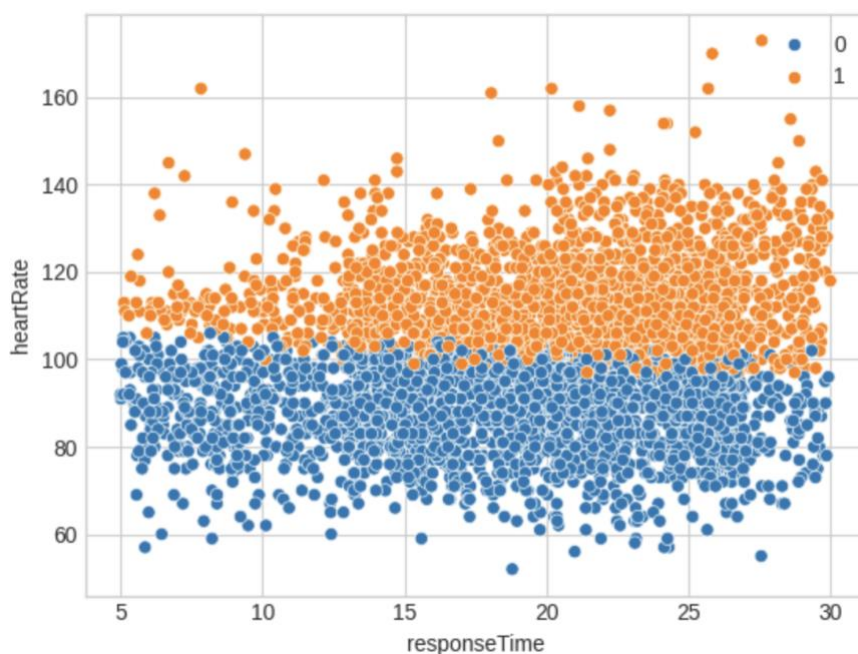


Рис. 3.6. Розподіл на два кластери

Таким чином ми бачимо, що успішне виконання переважає у низьких межах пульсу, здебільшого до 100, що утворює один кластер. Інший кластер утворюється вище 100. Час відповіді розподілений по двох кластерах так що виділити якісь деталі не виходить. Варто зазначити, що такі є певні аномальні значення коли пульс дуже високий, але відповідь вірна, та навпаки. Розподіл на два вектори зображений на Рис. 3.6.

Розбиття на два кластери для нашого датасету здається логічним, але таки варто провести визначення оптимальної кількості кластерів. Для цього проведемо навчання та визначимо оптимальне значення. Графік розподілу кластерів зображено на Рис. 3.7. Оптимальне значення кількості кластерів набуває коли функція набуває мінімального значення на осі y, а вісь x вказує на кількість кластерів. Оптимальне значення ми спостерігаємо у точці, де вісь x має значення 3. Збільшуючи кількість кластерів ми можемо отримати елементи перенавчання. У нашому випадку ми ризикуємо отримати кластери де всі дані сильно повторюються і навчання на цих даних буде мати високу точність, можливо навіть 1.00, але не буде мати ніякого практичного чи наукового значення так як будь-які дані з іншого кластеру будуть давати не вірне значення.

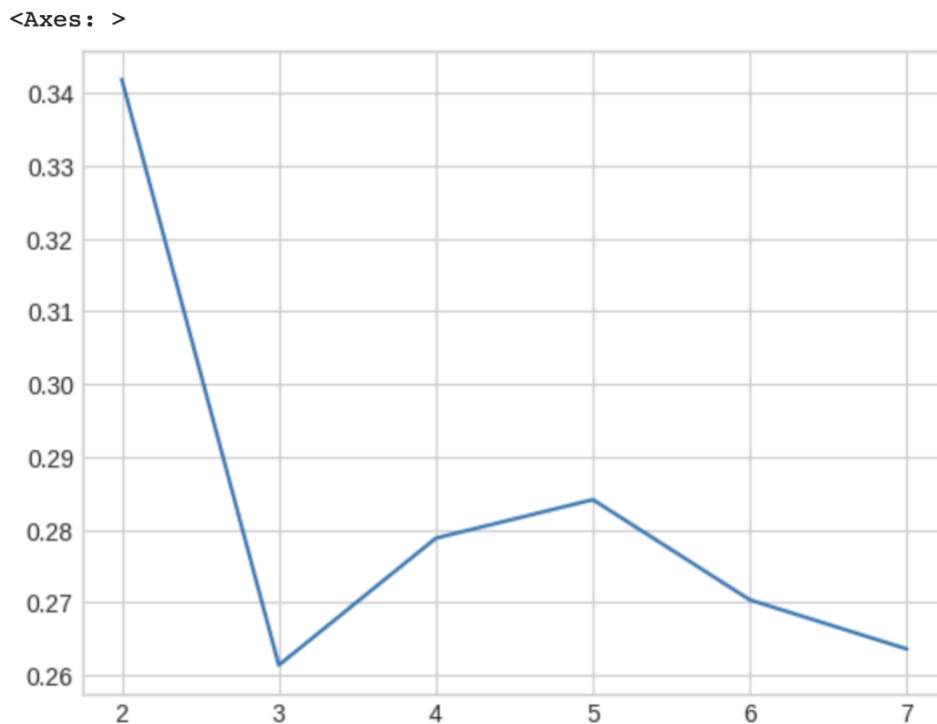


Рис. 3.7. Графік визначення кількості кластерів

Побудуємо розбиття для трьох кластерів, зображений на Рис. 3.8. Отримаємо поділ на три кластери, що поділяє стан людини-оператора у різних умовах, коли зростає пульс. Варто зазначити, що кластери 0 і 2 є бажаними, а кластер 1 – ні. Кластер 1 не є бажаним, бо значення пульсу в цьому кластері мають найвищі значення серед усіх трьох кластерів, що вказує на високий рівень стресу в людини-оператора.

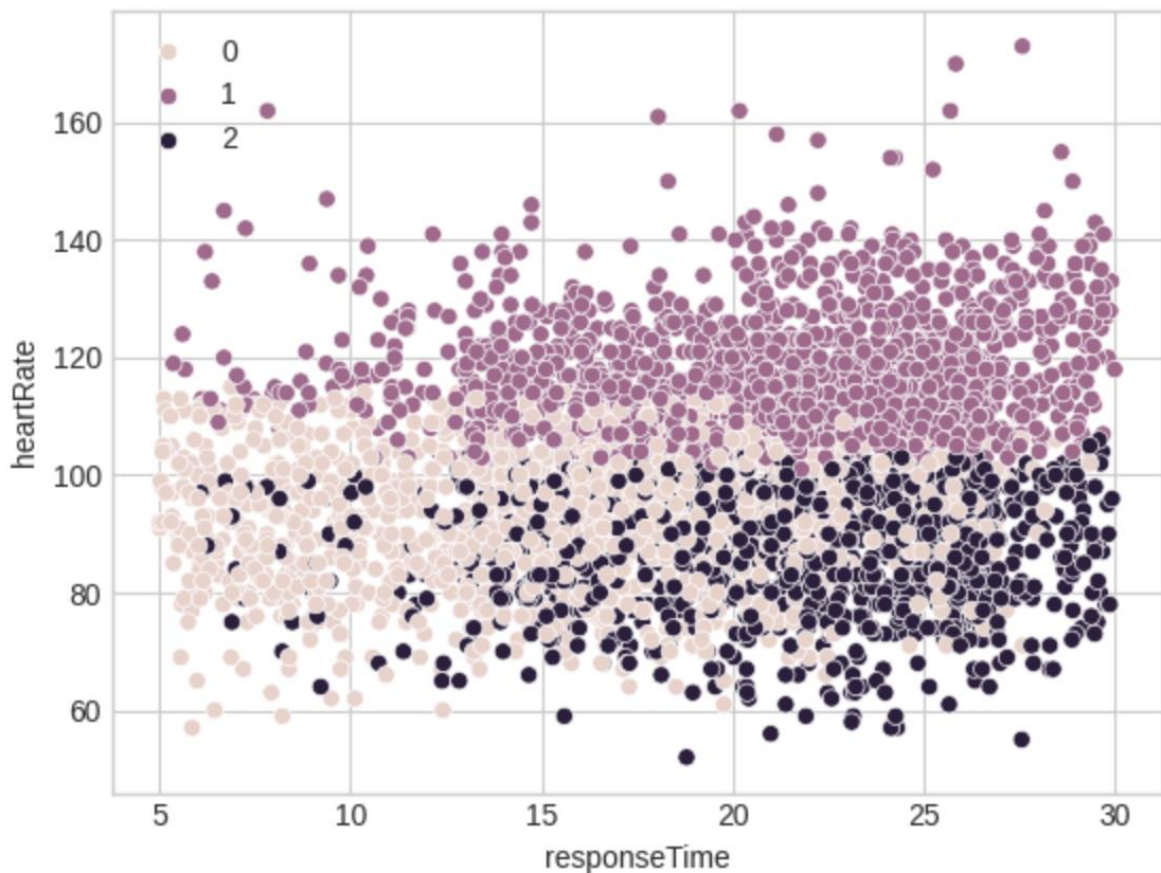


Рис. 3.8. Розподіл на три кластери

Результати кластерного аналізу методом К-середніх використаємо для подальшого удосконалення методу ієрархічного класифікатора. Поділ на три кластери використовуючи усі параметри оператора та не беручи до уваги індивідуальність оператора, а лише результати кожного окремого тесту як окремий результат показав кращі результати при розподілі на кластери методом К-середніх, а не ієрархічним кластерним аналізом. Для розподілу саме операторів на кластери використано ієрархічний кластерний аналіз у наступному підрозділі.

### 3.2.2. Ієрархічний кластерний аналіз

Ієрархічний кластерний аналіз (ІКА) – це метод кластеризації даних, який дозволяє групувати об'єкти в ієрархічні структури або дерева, де кожен вузол представляє кластер [66]. Цей метод дозволяє визначити структуру подібності об'єктів на різних рівнях деталізації, починаючи від окремих об'єктів і закінчуючи глобальними групами. ІКА ділиться на два головних підтипи: агломеративний і ділянковий аналіз.

Основні кроки ієрархічного кластерного аналізу:

- 1) *Початкова точка.* Починаючи зі всіх об'єктів як окремих кластерів, на першому кроці кожен об'єкт розглядається як окремий кластер.
- 2) *Обчислення подібності.* Обчислюються відстані або міри подібності між кластерами. Це може включати в себе використання метрик, таких як відстань Юкліда, кореляція, косинусна схожість і т.д.
- 3) *Об'єднання або розділення кластерів.* На кожному кроці два найбільш схожі кластера можуть бути об'єднані у великий кластер (агломеративний аналіз) або великий кластер розділений на менші підгрупи (ділянковий аналіз). Вибір способу залежить від методу, який ви використовуєте [67].
- 4) *Побудова ієрархії.* Процес об'єднання або розділення кластерів повторюється, поки всі об'єкти не належать до одного кластера на вершині ієрархії. Це створює ієрархічну структуру, де вищий рівень представляє групи з більшим рівнем загальних характеристик, а нижчий рівень - деталізовані кластери.
- 5) *Візуалізація і аналіз результатів.* Результат ієрархічного кластерного аналізу може бути візуалізований у вигляді дерева, де вершини представляють кластери на різних рівнях. Це дозволяє вам аналізувати структуру подібності та ієрархічні відносини між групами даних.

Переваги ієрархічного кластерного аналізу включають:

- Здатність визначити структуру подібності на різних рівнях деталізації.
- Не потребує попереднього визначення кількості кластерів.
- Можливість вивчати внутрішні зв'язки між групами об'єктів.

Недоліки ієрархічного кластерного аналізу включають:

- Вимогу до обчислювальних ресурсів, особливо при роботі з великими наборами даних.
- Часову складність при обробці великої кількості об'єктів.
- Потребу в виборі підходящого методу об'єднання чи розділення кластерів.

Ієрархічний кластерний аналіз знаходить застосування в різних галузях, таких як біологія, медицина, соціальні науки, маркетинг, комп'ютерна графіка і інші, де вивчення структури та відносин між даними має велике значення.

Проведемо кластерний аналіз за допомогою методу Ієрархічного Кластерного Аналізу для операторів. Першим кроком є визначення параметрів оператора, що будуть включені до кластерного аналізу. Побудуємо розподіл часу роботи оператора та визначимо тип розподілу. На Рис. 3.1. зображено поліноміальний розподіл часу роботи з тестом для одного з операторів, з якого отримуємо параметри оператора  $A = -0.4033$  та  $B = 6.2028$ . Ці параметри  $A$  та  $B$  є унікальними та відрізняються для кожного оператора, бо вони виведені з функції розподілу часу роботи оператора над тестом.

З часу також визначимо відношення різниці точок  $w_1$  та  $w_2$  до значення моди. Точки  $w_1$  та  $w_2$  фіксуються на значенні 0.5 моди.



Рис. 3.9. Поліноміальний розподіл часу роботи оператора

Фрактальний аналіз – це метод вивчення складних структур або явищ, які мають самоподібні властивості на різних масштабах. Для визначення фрактальної розмірності використаємо модифікований метод обчислення

фрактальної розмірності з визначенням співвідношення заповненості клітинок [68]. Також врахуємо показник Герста як  $H = 2 - D$ . Для визначення фрактальної розмірності та показника Герста візьмемо значення пульсу.

Використаємо наступні параметри оператора для кластерного:

- $P1$  – середнє значення пульсу в тестах I рівня складності,
- $P2$  – середнє значення пульсу в тестах II рівня складності,
- $P3$  – середнє значення пульсу в тестах III рівня складності,
- $T1$  – середній час виконання завдання в тестах I рівня складності,
- $T2$  – середній час виконання завдання в тестах II рівня складності,
- $T3$  – середній час виконання завдання в тестах III рівня складності,
- $W$  – відношення значення моди до довжини розподілу значень на проміжку значень 0.5 моди,
- $A$  – параметр  $A$  взятий з поліноміального розподілу,
- $B$  – параметр  $B$  взятий з поліноміального розподілу,
- $D$  – фрактальна розмірність,
- $H$  – показник Герста.

Отже, дістав подальший розвиток метод аналізу стану оператора, який відрізняється додаванням параметру Герста як функції від фрактальної розмірності, і який використовується в подальшому для кластеризації.

Кластерний аналіз операторів дасть можливість виділити певних операторів до однієї категорії. Проведемо кілька поділів операторів на кластери, щоб зрозуміти як вони мігрували між кластерами під час виконання завдань різного рівня складності, а пізніше побачимо результати аналізу по всіх параметрах.

При кластеризації важливи кроком є визначення оптимальної кількості кластерів. Оптимальна кількість кластерів дає можливість чітко виокремити операторів за їхніми параметрами до певного кластеру. Надлишкова кількість кластерів буде показувати майже завжди поокремих осіб, а в протилежній ситуації навпаки будуть вибрані в одному кластері оператори, що мають суттєво інші значення пульсу чи часу виконання. Тому для знаходження оптимальної кількості кластерів використаємо метод заснований на обчисленні

внутрішньокластерної суми квадратичних помилок (WSS) для різної кількості кластерів ( $k$ ) і виборі  $k$ , для якого зміна WSS спочатку починає зменшуватися. Ідея методу ліктя полягає в тому, що пояснена варіація швидко змінюється для невеликої кількості кластерів, а потім сповільнюється, що призводить до формування ліктя на кривій. Точка ліктя – це кількість кластерів, які ми можемо використовувати для нашого алгоритму кластеризації.

На Рис. 3.10. зображено дендрограму для параметрів P1, T1. Точка ліктя знаходиться на рівні 4.8, де можна виділити шість кластерів.

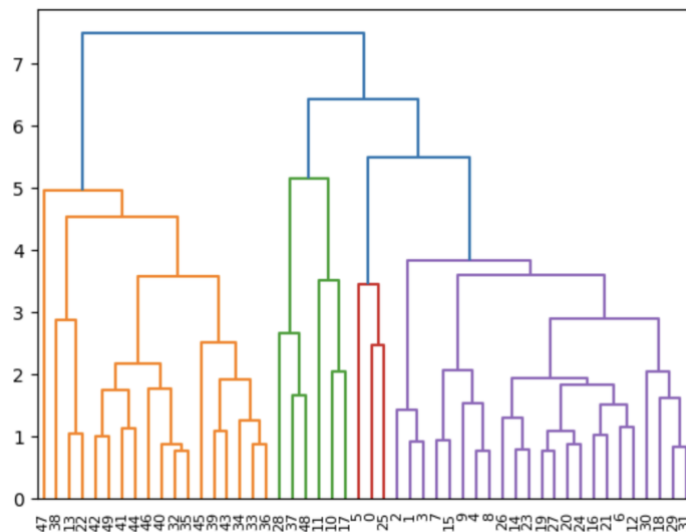


Рис. 3.10. Дендрограма кластеризації за параметрами першого рівня

Рис. 3.11. показує дендрограму для параметрів P2, T2 на рівні 4.5 також присутні шість кластерів. Кластеризації за параметрами P3, T3 показано на Рис 3.12. і присутні 5 кластерів на рівні 4.8.

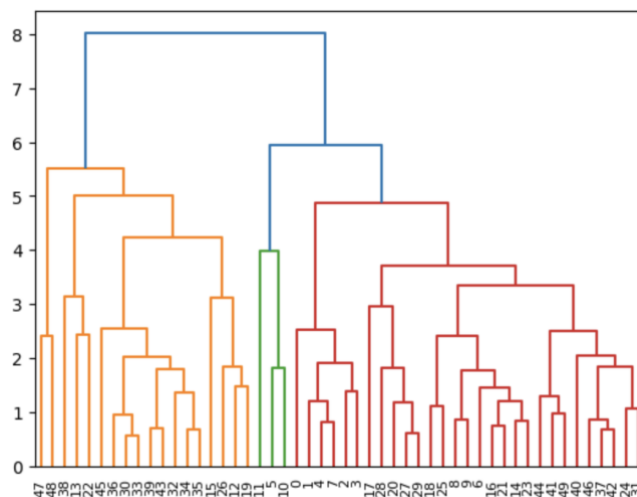


Рис. 3.11. Дендрограма кластеризації за параметрами другого рівня



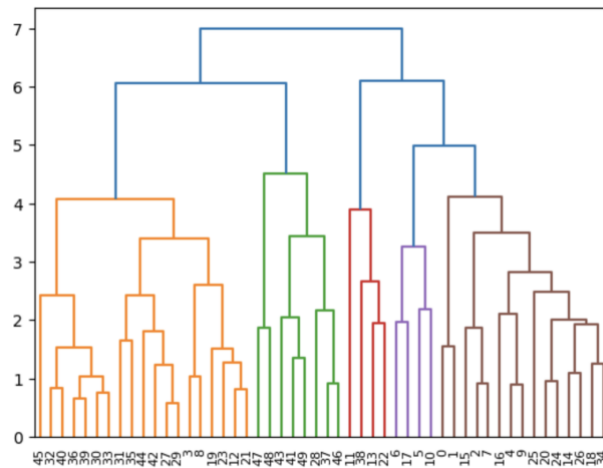


Рис. 3.12. Дендрограма кластеризації за параметрами третього рівня

Загалом оператори переходили між кластерами залежно від рівня складності тесту. За цим можна зрозуміти з якою складністю оператор може краще справлятися. Побудуємо дендрограму використовуючи усі параметри, зображено на Рис. 3.13 На рівні 5.0 маємо сім кластери:

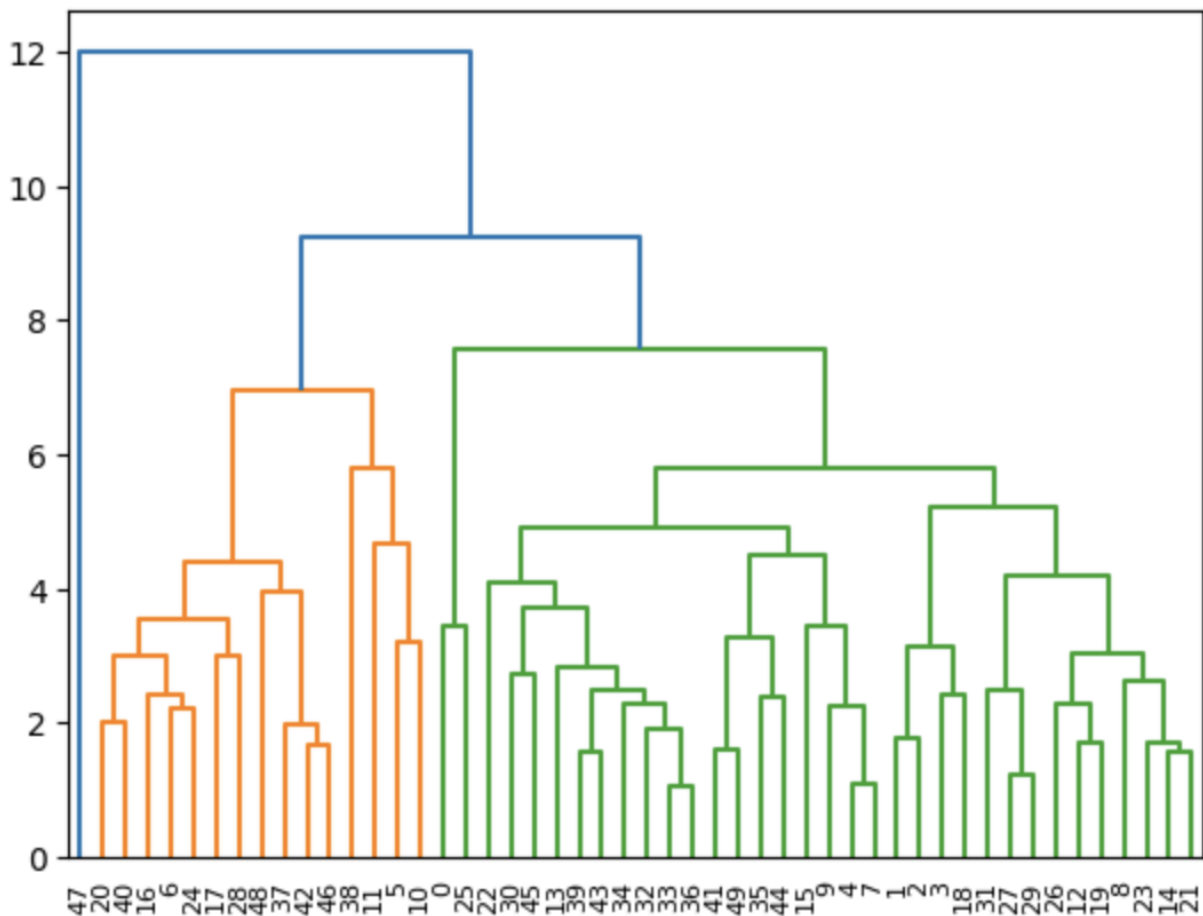


Рис. 3.13. Дендрограма кластеризації операторів

- 1 кластер має лише одного оператора, що виділився високим пульсом та високою точністю виконання тестів;
- 2 кластер включає в себе 11 операторів, що витратили більше часу на проходження тестів та більший відсоток успішного проходження відносно середнього значення;
- 3 кластер є ще одним представником одиночного кластеру, куди увійшов лише 1 оператор, який потребує мало часу для виконання тесту і має суттєво вищий пульс і успішність виконання тестів;
- 4 кластер включає 3 операторів з нижчим значенням пульсу і часу виконання, проте успішність нижча середньої;
- 5 кластер відзначився дуже низьким значенням пульсу, суттєво нижче середнього, але успішність дуже низька;
- 6 кластер має 18 операторів, відзначився найнижчою успішністю та середніми значеннями пульсу і часу виконання;
- 7 кластер включає 14 операторів, відзначається серединою успішністю та нижчими значеннями пульсу.

Провівши кластерний аналіз по операторах з параметрами з різних рівнів, ми отримуємо характеристику та розуміння як оператори справляються виконуючи завдання різних рівнів. У результаті отримуємо метод за яким можемо визначати, які оператори краще справляються з роботою різного рівня та можемо надавати рекомендації при наборі кадрів.

Також результати кластерного аналізу – використані при подальшому аналізі з обрахунками багатовимірної середньої, ієрархічному класифікаторі.

### **3.3. Аномальні значення та закономірності**

Результати методів класифікації показали результати з точністю більше 80% і менше 90% то ж має сенс провести чистку аномалій у наборі даних та провести повторні дослідження. Аномалії у датасеті – це несподівані та незвичайні значення або патерни даних, які відхиляються від звичних, типових або очікуваних зразків [62]. Аномалії можуть бути результатом помилок вводу даних,

аномальних подій, випадкових відхилень, специфічних подій чи надзвичайних ситуацій. Пошук аномалій може мати багато цілей: пошук пропущених даних, потенційні шахрайства якщо це фінансова сфера, відслідковування якості виробництва чи моніторинг ситуації, що вимагають оперативного реагування. Загалом, пошук аномалій у датасеті сприяє покращенню якості даних, безпеці та ефективності процесів, а також допомагає приймати обґрунтовані та своєчасні рішення на основі інформації.

На противагу аномальним значенням йде закономірність. Закономірності показують зв'язок та залежність між послідовними чи навіть на перший погляд не пов'язаними подіями. Пошук закономірностей в датасеті – це процес виявлення структури, залежностей та патернів в даних. Знайдені закономірності можуть бути використані для побудови прогностичних моделей та прогнозу результатів на основі вхідних даних. Наприклад, у машинному навчанні, пошук закономірностей допомагає створити точні моделі. Закономірності допомагають виділити нормальний стан даних, і тим самим, виявити аномалії або незвичайні події, які відхиляються від цих закономірностей. У підсумку, пошук закономірностей в датасеті сприяє розумінню, використанню та оптимізації даних, що робить його важливим етапом в аналізі та використанні інформації з наборів даних.

Таблиця 3.2. Результати проходження тестів.

Номер оператора	Вік	Середній час усіх тестів, с	Середній час по правильних тестах, с	Середній час по помилкових тестах, с	Кількість правильних тестів	Кількість помилкових тестів
1	23	19.619	19.415	20.097	84	36
2	25	20.256	20.525	19.516	88	32
3	27	19.793	19.511	20.639	90	30
...	...	...	...	...	...	...
50	63	19.353	18.851	20.525	84	36

На основі результатів тестування людини-оператора в таблиці 3.2. приведені такі дані щодо часу виконання завдань для кожного індивідуума, доповнено в додатку Б.2. З наданої інформації можна визначити, що час реакції та точність виконання тесту мають індивідуальний характер, і відхилення в обох напрямках виявляються у різних ступенях. Результати тестів для усіх операторів відображені у таблиці 3.2.

Наступним етапом розгляду є вивчення відношень між успішним та неуспішним проходженням тестів в зазначеному експерименті, результати наведені в таблиці 3.3.

Таблиця 3.3. Розподіл пульсу по виконанню тестів.

	Успішне виконання «yes»	Неуспішне виконання «no»
Пульс mean	94.8	114.4
Пульс max	156	173
Пульс min	51	57
Час mean	19.28	19.32

При успішному виконанні завдань у людини-оператора середній пульс складає 94.8 уд/хв, тоді як при неуспішному - 114.4 уд/хв. Це свідчить про вагому взаємозалежність між виконанням завдань та пульсом, що підтверджує високий рівень стресу, який впливає на здатність концентрації та точності виконання завдань. Середні показники часу, витраченого на успішне та неуспішне виконання, становлять відповідно 19.32 та 19.28 секунд. Хоча невелика різниця свідчить про те, що людина-оператор може витратити більше часу на розгляд чи пошук об'єкта в рамках завдання, результат викликає увагу через мінімальну різницю у 0.04 секунди, яка є надто невеликою в контексті власного значення.

Розглянемо параметри оператора провівши обчислення методом багатовимірної середньої по кластерах. У результаті ієрархічної кластеризації на рівні 5.0 було виділено 7 кластерів і до них увійшли відповідні оператори:

- Кластер 1: 48;
- Кластер 2: 21, 41, 17, 7, 26, 18, 29, 49, 38, 43, 47;
- Кластер 3: 39;

- Кластер 4: 12, 6, 11;
- Кластер 5: 1, 26;
- Кластер 6: 23, 31, 46, 14, 40, 44, 35, 33, 34, 37, 42, 50, 36, 45, 16, 10, 5, 8;
- Кластер 7: 2, 3, 4, 19, 32, 28, 30, 27, 13, 20, 9, 24, 15, 22.

Це метод, що передбачає знаходження середнього значення серед ознак, у випадку з операторами(кластерами операторів) їх параметрів. Після цього обчислюється значення нормованих характеристик – відношення параметрів кластера до середнього значення по кластерах. Беручи значення нормативних характеристик обчислюється значення багатовимірної середньої та присвоюються ранги. Провівши обчислення отримуємо наступні результати в таблиці 3.4.

Таблиця 3.4. Результат знаходження багатовимірної середньої.

Кластер	Характеристики ознак		Нормовані характеристики		Багато-вимірна середня	Ранги
	Час виконання	Пульс	Час виконання	Пульс		
1	19,08	135,47	1,001	1,307	1,154	7
2	19,66	105,43	1,031	1,017	1,024	6
3	18,1	108,1	0,95	1,043	0,996	5
4	18,86	98,55	0,99	0,951	0,970	2
5	19,23	82,48	1,008	0,796	0,902	1
6	19,08	99,36	1,001	0,959	0,98	4
7	19,43	95,97	1,02	0,926	0,973	3
Середнє	19,06	103,62				

Значення багатовимірної середньої поділило кластери операторів за рангами від 1 до 7. Поділимо операторів за рангом на три категорії: 1 – значення багатовимірної середньої до 0.95; 2 – значення до 1.0; 3 – значення 1.0 та вище. До першої категорії віднесемо операторів з кластеру 1 з рангом 1. До наступної категорії увійшли оператори з 4 кластеру з рангом 2, з 7 кластеру з рангом 3, з 6

кластеру з рангом 4 та з кластеру 3 з рангом 5. До третьої категорії увійшли оператори з кластеру 2 з рангом 6 та з 1 кластеру з рангом 7. Таким чином отримуємо групи кластерів операторів за їх параметрами.

Перед пошуком аномалій побудуємо графік зміни пульсу від порядкового номеру тесту, тобто за зростаючою складністю, зображено на Рис. 3.14. Графік показує поступове зростання пульсу, але є кілька аномальних ситуацій, де середнє значення пульсу вибивається з попереднього та наступного. Це говорить, що цей або попередній тест був легший.

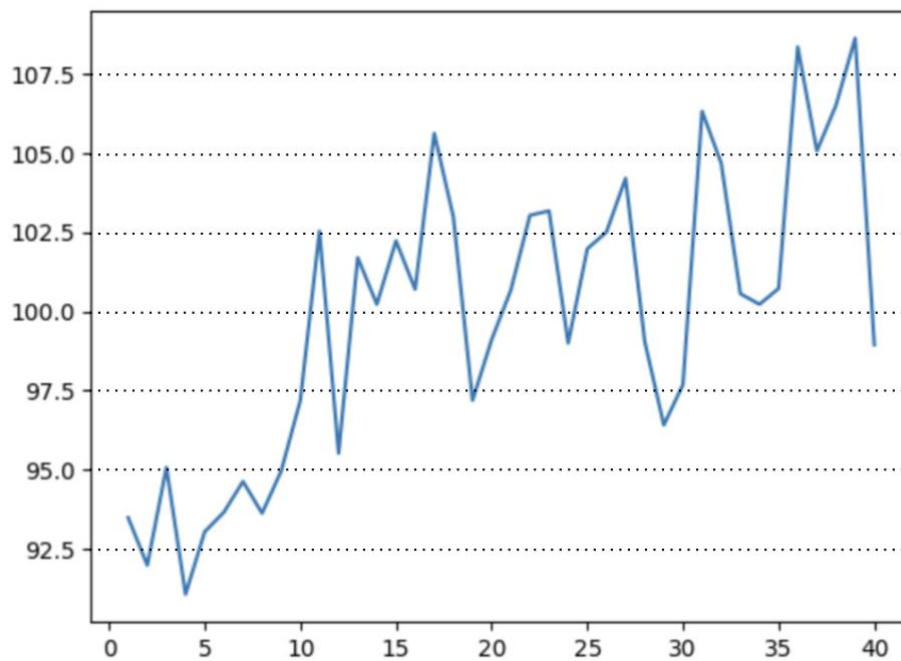


Рис. 3.14. Графік залежності середнього пульсу від тесту

Порівнюючи значення наступного кола тестування можна зрозуміти як змінюється пульс людини-оператора виконуючи тест по групі складності. Ці дані є важливими при виконанні тесту в довільному порядку так як зміна пульсу може відбуватися не лінійно, бо тести попадаються не в підвищенні рівня складності, а випадково. Також зворотній момент, що може впливати втима на користувача і вкінці він вирішує складні завдання довго і нервується та зростає пульс, а якщо такі завдання йдуть на початку то він може не хвилюватись і пульс буде нижчий ніж середні значення.

Наступним етапом є пошук випадків, де людина відволіклась на подразники та затягнула з тим, що дати відповідь на завдання. Кожна людина-оператор

проходила той самий набір тестів тричі з перервою від кількох днів до кількох тижнів. Якщо оператор виконав тест №1 за 6 і 7 секунд перші два рази, а на третій більше 25 то ймовірно щось його відволікло. Іншим чинником ми маємо зміни пульсу, оскільки кожен раз в людини інший психологічний стан, вона по різному налаштована то зміни пульсу будуть про це говорити.

У результатах тестування маємо багато випадків, коли ми можемо спостерігати такі аномальні значення, наведемо лише кілька прикладів. Тест №39 людина-оператор виконала 3 рази в межах (24, 27) секунд, двічі пульс був низьким до 73, а одного разу 112, це суттєвий стрибок і в цей момент оператор зробив помилку. Наступний приклад тест №7 людина мала пульс в межах (71, 80), що відповідає стану спокою, але перший раз виконала тест за 20 секунд, другий за 12, а на третій відповіла на останній секунді. У результаті детального аналізу ми бачимо такі ситуації, також бувають зворотні ситуації, що оператор поспішив з відповіддю, або був у стані спокою при важкому тесті і дав відповідь правильно. Але такий пошук аномалій дає можливість розуміти залежності між успішністю виконання завдань та станом оператора і його навколишнього середовища.



Рис. 3.15. Графік залежності середнього пульсу від тесту після фільтрації

Виключимо зі списку ці тести та побудуємо графік знову на Рис. 3.15. На ньому видно, що пульс зростає рівномірно, окрім одного моменту, але це група тестів, а не лише один випадок, що понизив пульс. Таким чином ми можемо відокремити легші тести і подати їх в іншій послідовності або ж зовсім прибрати.

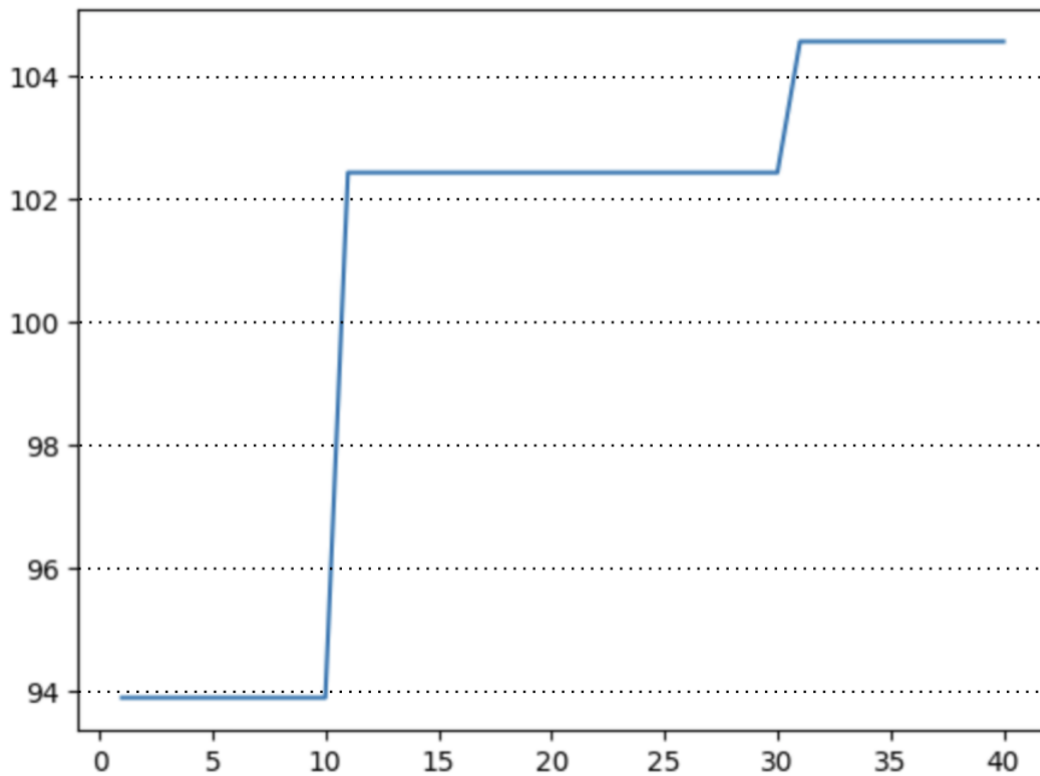


Рис. 3.16. Графік залежності середнього пульсу по рівнях складності

На Рис 3.16. зображено графік залежності пульсу по різних рівнях складності, що дає можливість зрозуміти наскільки пульс вибивається від середнього значення в окремих випадках.

Провівши очищення даних, вилучимо аномалії та проведемо навчання методом xgboost та метод побудови дерева рішень ще раз.

Таблиця 3.5. Результат навчання по тестах.

Метод	Набір даних	F1 Score
XGBoost	Без попередньої обробки	90%
Decision Tree	Без попередньої обробки	85%
XGBoost	З попередньою обробкою	90.4%



Decision Tree	З попередньою обробкою	85.7%
---------------	------------------------	-------

F1 score є важливою метрикою у задачах класифікації, особливо коли у вас є дисбаланс класів в наборі даних. Використання F1 score є особливо корисним у випадках, коли вам потрібно узагальнити ефективність алгоритму не лише за точністю (accuracy), але й за умови того, що важливо врахувати як false positives, так і false negatives.

Отримані дані по F1 score показують незначно, але кращі результати. Наступним кроком буде провести кластеризацію та визначити результати точності по різних кластерах, що має підвищити точність.

### 3.4. Метод ієрархічної класифікації для визначення рівня стресостійкості

На основі побудованої східцевої функції для кожного оператора отримаємо модель поведінки стану особи. Далі розробимо ієрархічний класифікатор на основі ансамблю та динамічного зважування для визначення рівня стресостійкості.

Ієрархічний класифікатор (або ієрархічна модель) є одним з підходів до розв'язання завдань класифікації в області data science, де об'єкти поділяються на кілька рівнів ієрархії або категорій. Цей підхід використовується тоді, коли маємо справу з великою кількістю класів, які можна організувати в ієрархічну структуру для полегшення класифікації.

Він побудований на основі двох кроків:

- Ієрархічна кластеризація;
- Класифікація в межах кожного кластеру;
- Зважування результатів.

Дендрограма – це графічне представлення результатів ієрархічної кластеризації [69]. Ієрархічна кластеризація – це метод групування об'єктів або спостережень в ієрархічну структуру. Дендрограма використовується для візуалізації цієї ієрархії та спрощення аналізу структури кластерів. Основні етапи використання дендрограм у кластеризації даних:

- Вибір метрики відстані. Першим кроком є вибір метрики відстані, яка визначає відстань між двома об'єктами чи кластерами. Зазвичай використовуються метрики, такі як евклідова відстань, Манхеттенська відстань або кореляційна відстань, залежно від природи даних [70].
- Визначення матриці відстаней. Обчислюється матриця відстаней між всіма об'єктами чи спостереженнями. Ця матриця служить основою для побудови ієрархічної структури.
- Побудова дендрограми. На основі матриці відстаней побудована дендрограма, яка представляє собою дерево, де кожен вузол представляє об'єкт чи кластер, а гілки вказують на відстань між ними. Початково кожен об'єкт вважається окремим кластером, і далі об'єднуються в єдині кластери, поки не сформується повна ієрархія [71].
- Вибір оптимального кількості кластерів. Аналізуючи дендрограму, можна визначити оптимальну кількість кластерів. Це може бути здійснено, шукаючи великі відстані між гілками дерева, які можна вважати межами між кластерами.
- Розділення на кластери. За оптимальною кількістю кластерів можна використовувати алгоритми відсічення (cutting) дендрограми для отримання фактичних кластерів.

Дендрограми допомагають візуалізувати структуру ієрархічної кластеризації та роблять процес вибору кількості та розміщення кластерів більш інтуїтивно зрозумілим. Це особливо корисно при великих наборах даних або у випадках, коли хочете розуміти взаємозв'язки між кластерами та їхньою ієрархією.

У багатьох реальних програмах, особливо тих, які генерують великі обсяги даних, дані часто стають доступними пакетами протягом певного періоду часу. Ці програми потребують механізму для поступового включення додаткових даних у базу знань, бажано без доступу до попередніх даних [72]. Формально кажучи, поступове навчання означає послідовне оновлення гіпотези з використанням поточних даних і попередніх гіпотез, але не попередніх даних, таким чином, що поточна гіпотеза описує всі дані, які були отримані до цього

часу. Поступове навчання пов'язане з добре відомою дилемою стабільність-пластичність, де стабільність означає здатність алгоритму зберігати наявні знання, а пластичність – здатність алгоритму отримувати нові дані. Удосконалення одного зазвичай відбувається за рахунок іншого. Наприклад, онлайн-алгоритми потокової передачі даних зазвичай мають добру пластичність, але низьку стабільність, тоді як багато добре відомих керованих алгоритмів, таких як MLP, SVM і kNN, мають добру стабільність, але погані властивості пластичності.

Системи на основі ансамблів забезпечують інтуїтивно зрозумілий підхід до поступового навчання, який також забезпечує збалансоване рішення дилеми стабільності та пластичності. Розглянемо алгоритм AdaBoost, який спрямовує наступні класифікатори до все більш складних випадків. У налаштуваннях інкрементального навчання деякі з екземплярів, представлених новою партією, також можна інтерпретувати як «складні», якщо вони несуть нову інформацію. Тому підхід, подібний до AdaBoost, можна використовувати в налаштуваннях поступового навчання з певними модифікаціями, такими як створення нового ансамблю з кожною доступною партією; скидання розподілу вибірки на основі продуктивності існуючого ансамблю на новому пакеті даних навчання та пом'якшення пункту переривання. Зауважте, однак, що правило оновлення розподілу в AdaBoost спрямовує розподіл вибірки на ті екземпляри, неправильно класифіковані попереднім класифікатором. У налаштуваннях інкрементального навчання необхідно скерувати алгоритм, щоб він зосередився на тих нових екземплярах, введених новою партією даних, які ще не вивчені поточним ансамблем, а не попереднім класифікатором [73, 74].

Проблема поступового навчання стає особливо складною, якщо нові дані також вводять нові класи. Це пояснюється тим, що класифікатори, попередньо навчені на попередніх пакетах даних, неминуче неправильно класифікують екземпляри нового класу, на якому вони не були навчені. Тільки нові класифікатори здатні розпізнавати новий клас(и). Таким чином, будь-яке рішення нових класифікаторів щодо правильного вибору нового класу перевершується попередніми класифікаторами, поки не буде достатньо нових класифікаторів,

щоб протистояти загальному голосуванню цих початкових класифікаторів. Отже, потрібна відносно велика кількість нових класифікаторів, які розпізнають новий клас, щоб їх загальна вага могла перезаписати неправильні голоси вихідних класифікаторів.

Метод DW-CAV Динамічне зважування – консультація та голосування (Dynamically Weighed – Consult and Vote) [75]. Вхідні дані: екземпляр  $x_i$  для класифікації, усі попередньо створені класифікатори  $h_t^k$ , нормалізовані значення помилок  $\beta_t^k$ , мітки класів  $CL_t^k$  використані в навчанні  $h_t^k$ . Ініціалізація ваг класифікатора:  $W_t^k = \log(1/\beta_t^k)$ . Розрахунок кожного  $\omega_c \in \{\omega_1, \dots, \omega_c\}$ .

1) Фактор нормалізації, наведений в формулі 3.2:

$$Z_c = \sum_k \sum_{t:c \in CL_t^k} W_t^k \quad (3.2)$$

2) Класова впевненість, показано в формулі 3.3:

$$P_c(i) = \frac{\sum_k \sum_{t:h_t^k(x_i)=\omega_c} W_t^k}{Z_c} \quad (3.3)$$

3) Якщо  $P_k(i) = P_l(i), k \neq l$  тоді  $\mathcal{E}_k \cap \mathcal{E}_l = \emptyset, P_k(i) = P_l(i) = 0$ , де  $\mathcal{E}_k$  – набір класифікаторів, що бачили клас  $\omega_k$ .

Оновлюємо значення ваг, формула оновлення 3.4, для екземпляру  $x_i$ :

$$W_t^k(i) = W_t^k \cdot \prod_{c:\omega_c \notin CL_t^k} (1 - P_c(i)) \quad (3.4)$$

Підсумкові обрахунки представлені в формулі 3.5.

$$H_{final}(x_i) = \arg \max_{\omega \in \Omega} \sum_k \sum_{t:h_t^k(x_i)=\omega_c} W_t^k(i) \quad (3.5)$$

Композитна гіпотеза  $H_t^k$  для перших  $t$  класифікаторів із  $k$ -ї партії обчислюється голосуванням зваженої більшості всіх класифікаторів із використанням вагових коефіцієнтів  $W_t^k$ , які самі зважені на основі достовірності кожного класу класифікаторів  $P_c$  ((3.2) і (3.3)). Специфічна впевненість класу  $P_c(i)$ , наприклад  $x_i$ , є відношенням загальної ваги всіх класифікаторів, які вибрали клас  $\omega_c$  (наприклад  $x_i$ ) до загальної ваги всіх класифікаторів, які бачили клас  $\omega_c$ .

Отже,  $P_c(i)$  представляє колективну впевненість класифікаторів, навчених розпізнавати клас  $\omega_c$ , у виборі класу  $\omega_c$ , наприклад  $x_i$ . Високе значення  $P_c(i)$ ,

близьке до 1, вказує на те, що класифікатори, навчені розпізнавати клас  $\omega_c$ , фактично переважно вибрали класу  $\omega_c$ , а тому ті, хто не пройшов навчання щодо  $\omega_c$ , не повинні голосувати (або зменшувати свою вагу голосування) для цього випадку [75].

Поділені кластери утворюють три групи параметрів оператора. Кожен кластер має однорідні дані, таким чином точність навчання в середині кластеру буде вищою. Однорідні дані в кластері вказують на те, що об'єкти або спостереження, які належать до одного кластера, мають подібні характеристики або властивості. Однорідність кластера вказує на те, що об'єкти всередині кластера схожі один на одного з певного погляду чи за певними властивостями. Провівши навчання XGBoost і методом побудови дерева рішень, отримуємо результати вищі ніж були на повному наборі даних. Приріст у F1 Score більший ніж після попередньої обробки даних. У кожному окремому кластері результат 92.5% і вищий для методу XGBoost, а для дерева рішень 87% і вище.

Таблиця 3.6. Результати навчання після кластеризації.

Метод	Набір даних	F1 Score	
XGBoost	Без попередньої обробки	90%	
Decision Tree	Без попередньої обробки	85%	
XGBoost	З попередньою обробкою	90.4%	
Decision Tree	З попередньою обробкою	85.7%	
Метод	Cluster 1 – F1 Score	Cluster 2 – F1 Score	Cluster 3 – F1 Score
XGBoost	94.6%	92.5%	94.5%
Decision Tree	92%	87%	90%

Поділені кластери мають більш однорідні дані, але перевіривши чи немає ефекту перенавчання видно, що вони поділені добре та ефект перенавчання відсутній.

Розглянемо детальніше дерево рішень для кластеру №1 на Рис. 3.17. Перевіривши значення гінї видно, що розподіл в даному кластері має переважно

значення коли людина-оператор пройшла тест успішно. Уже на першому кроці значення  $gini$  0.135, а розподіл правильних і неправильних відповідей 52 і 663, не дивно, що точність у цьому кластері найвища.

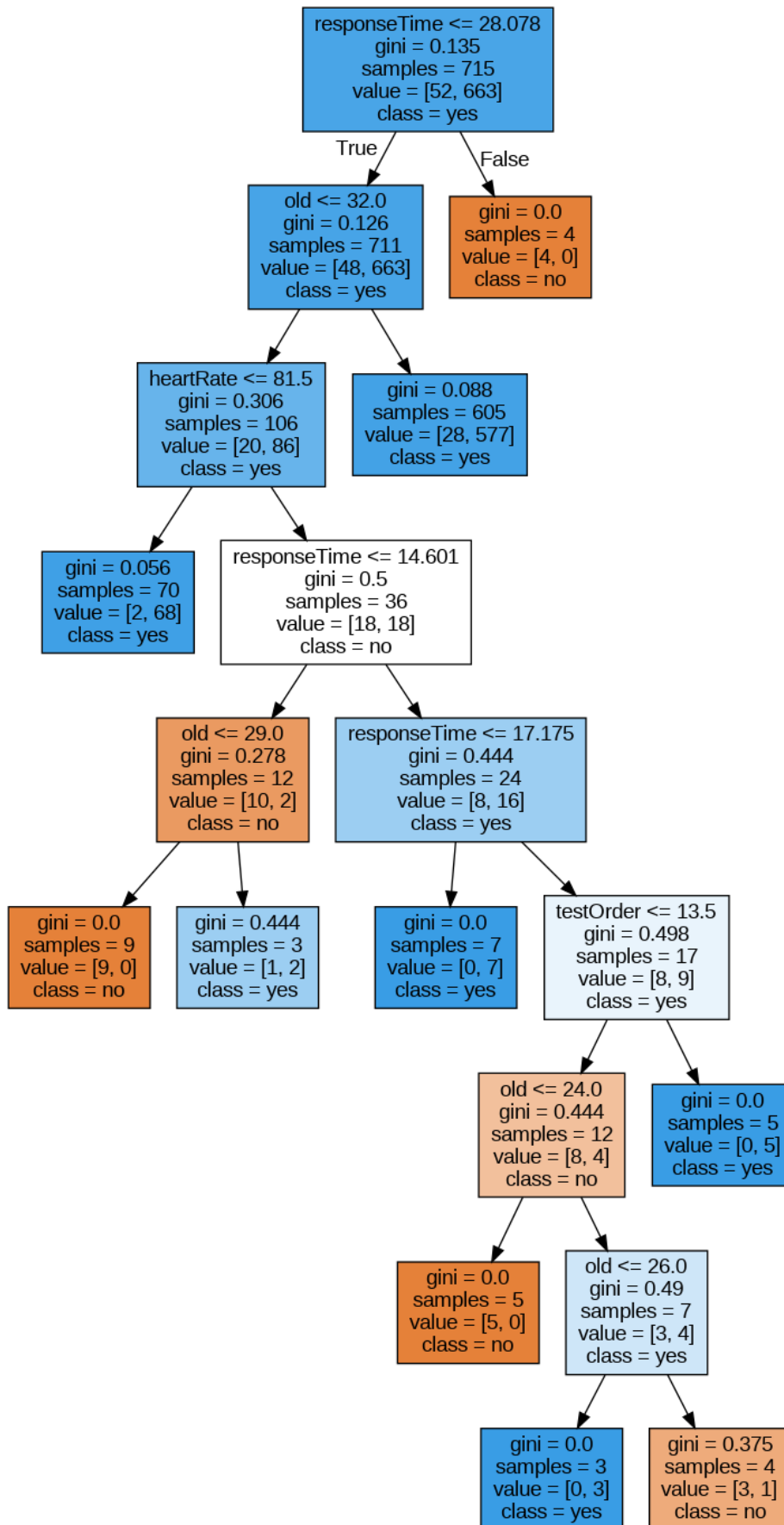


Рис. 3.17. Дерево рішень для Cluster №1

Розглянемо дерево рішень для кластеру №2 на Рис. 3.18. Тут значення  $gini$  0.49, що вказує на дуже рівномірний розподіл між значеннями. Кількість правильних і неправильних відповідей є 302 і 401, що також говорить про рівномірний розподіл. Більшість нащадків в дереві мають рівномірний розподіл або невелику кількість значень і низький розподіл, лише варто зазначити, що є один вузол куди попало 6 значень і всі вони одного класу. Точність в даному кластері 0.87 за методом дерева рішень та 0.925 за методом XGBoost, це нижча точність ніж у першому кластері, але вище ніж при аналізі всього набору даних.

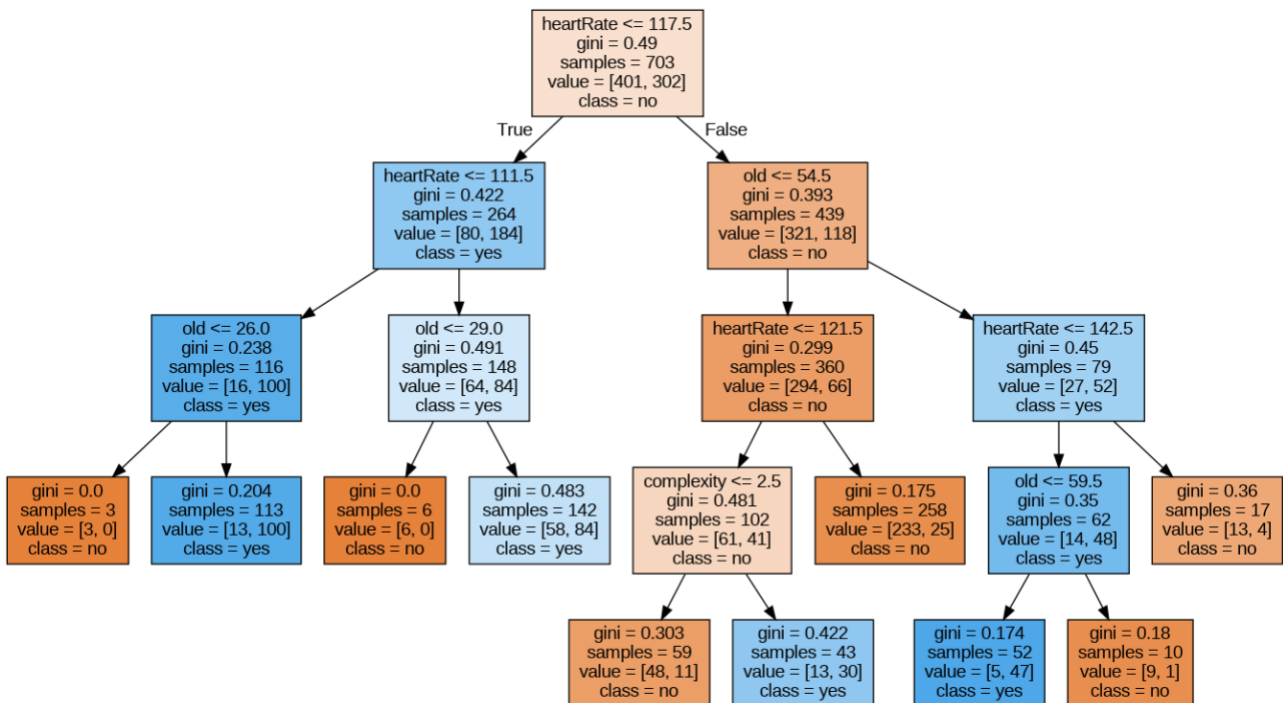


Рис. 3.18. Дерево рішень для Cluster №2

Розглянемо дерево рішень для кластеру №3 на Рис. 3.19. побудувавши дерево рішень видно, що кластер №3 по розподілу даних виглядає значно ближче до дерева по кластер №1. Тут значення  $gini$  0.172 в батьківському вузлі, але є нащадок, що має значення  $gini$  0.106 і 604 екземпляри. Цей вузол визначає значну кількість правильних відповідей. Інша ж частина дерева розподілена рівномірно, невеличкими підгрупами. Також точність навчання знаходиться між 1 і 2 кластером та є значно вищою ніж весь набір даних. З побудованого дерева також видно поділ важливості кожної з властивостей в конкретному кластері. До

третього кластеру попала більшість тестів з часом виконання 29,369 секунди і більше. Також межі порівняння пульсу є невисокими відносно іншого кластеру видно, що сюди попало менше стресових ситуацій.

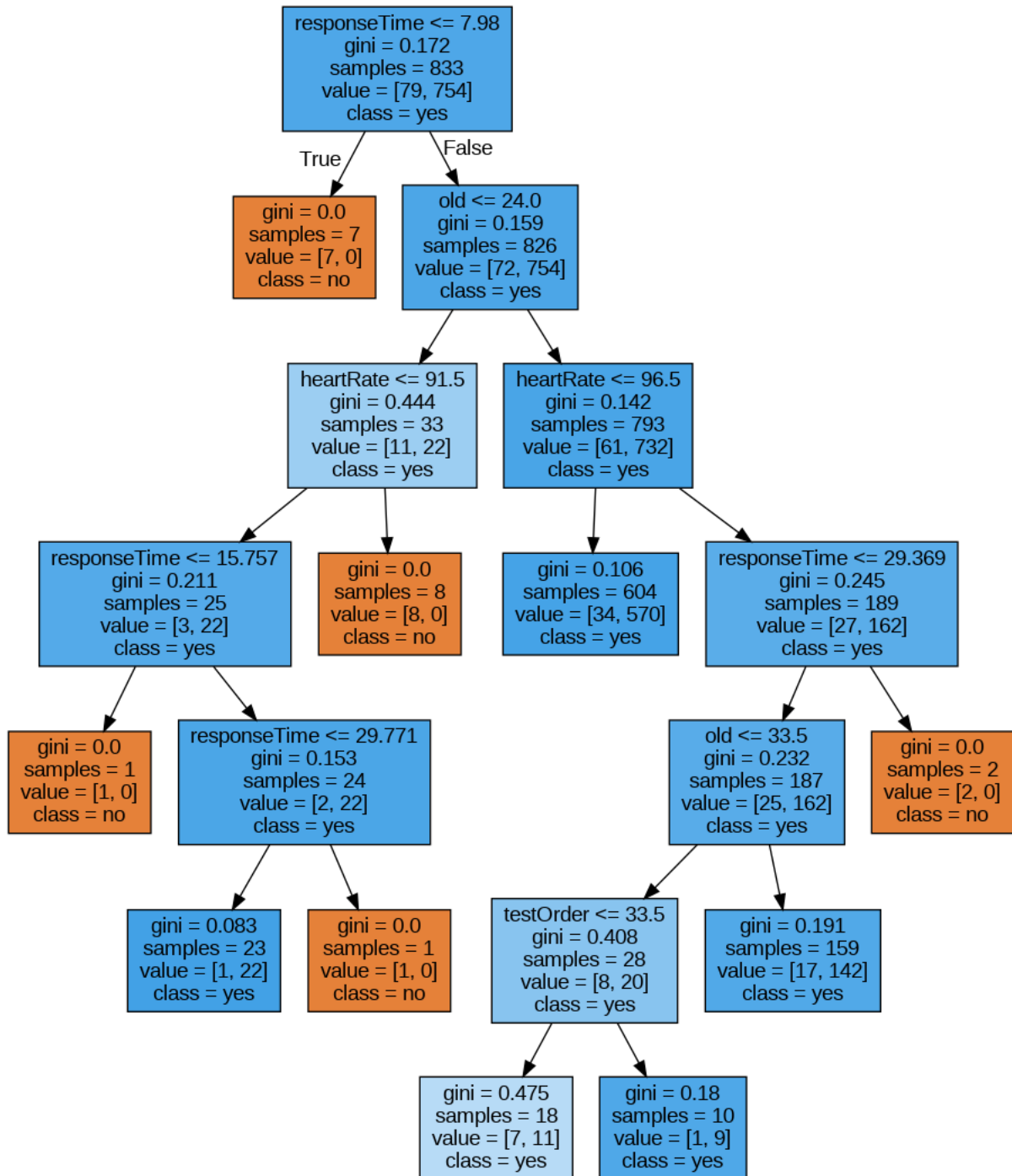


Рис. 3.19. Дерево рішень для Cluster №3

Отже, метод-ансамбль методів кластеризації та класифікації показав високі результати та успішно пройшов перевірку на наявність перенавчання за допомогою кросвалідації. Кросвалідація є методом оцінювання ефективності



моделі в машинному навчанні, включаючи задачі класифікації. Основна ідея полягає в тому, щоб розділити набір даних на декілька піднаборів, тренувати та тестувати модель кілька разів, змінюючи набір для тестування, і потім усереднити результати.

Такий ієрархічний класифікатор має дуже важливий крок вибору оптимальної кількості кластерів та перевірки даних у кластерах. Також побудова дерев по кластерах краще розкриває властивості всього набору даних.

### **Висновки до третього розділу**

Проведено кластеризацію операторів за їхніми параметрами по рівнях та загальну. Введено та використано такі параметри як фрактальна розмірність та показник Герста для кластеризації операторів.

Зроблено ранжування операторів по кластерах за допомогою методу багатовимірної середньої, що дало можливість погрупувати операторів та ефективно прогнозувати їхню ефективність. Проведено пошук аномалій у наборі даних та побудовано гістограми розподілу даних. Побудовано графіки залежності від різних параметрів оператора.

Також у даному розділі було розроблено метод-ансамбль класифікації для аналізу параметрів оператора з використанням комп'ютерних тренажерів. Метод-ансамбль було розроблено шляхом проведення навчання за допомогою методів класифікації даних. Отримані результати точності з самих методів класифікації були вищі 0.85, але поділ на кластери показав можливості до покращення точності на даних по кластерах.

Наведено дерева рішень для загального набору даних, що показує розподіл даних по дереву та вказують основні властивості, що покращують точність результатів. Також подано дерева рішень по окремих кластерах, де дані більш однорідні. Отримані результати по навчанню перевірені чи нема моменту перенавчання.

## РОЗДІЛ 4. РОЗРОБКА АРХІТЕКТУРИ ТА АПРОБАЦІЯ РЕЗУЛЬТАТІВ

У розділі розроблено інформаційну технологію для експрес оцінки параметрів операторської діяльності. Представлено схему зберігання даних, наведено архітектуру системи та її основних модулів, обґрунтовано технології та засоби для реалізації системи. Представлено функціонал системи, основні сценарії роботи з користувачем та адміністратором.

### 4.1. Структура сховища даних

При побудові архітектури систему важливим кроком є визначення даних, що будуть зберігатися, мігрувати та жити в системі. Також їхні типи, зв'язки та об'єми. Також важливим кроком є звідки система отримує ці дані. У випадку розробки системи комп'ютерного тренажера для людини-оператора та менеджерів по підбору персоналу є можливість отримування даних не лише від користувача та адміністратора, а й з використанням загальнодоступних даних чи використання різних сенсорів.

Мультиmodalні дані – це дані, які складаються з інформації різних modalностей або джерел, таких як текст, зображення, звук, відео, сенсорні дані тощо. Ці дані можуть містити інформацію з різних джерел або спостережень та охоплювати різні аспекти досліджуваного об'єкта або явища [76].

У сфері штучного інтелекту та обробки природної мови, мультиmodalні дані можуть використовуватися для створення багатоаспектних моделей, які можуть розпізнавати та розуміти текст, зображення та інші формати одночасно. Це особливо важливо в контексті розробки інтелектуальних систем, таких як системи розпізнавання мови, об'єктів на зображеннях, аналізу відео або впливу показників людини на її роботу.

Використання сенсорних даних з розумних гаджетів розширює можливості для аналізу впливу різних чинників на роботу людей, зокрема людини-оператора комп'ютерного тренажера. Більшість розумних гаджетів від смарт-браслету до розумного чи спортивного годинника мають можливість відслідковувати пульс. Дорожчі та новіші версії мають можливість робити ЕКГ серця, Apple Watch s8 як

приклад. Ці дані надходять з іншого джерела та можуть використовуватись з основними параметрами людини-оператора комп'ютерного тренажеру.

Для зберігання даних використаємо СКБД AWS RDS. Amazon Web Services (AWS) Relational Database Service (RDS) – це високоефективний та інтелектуально сконструйований сервіс для керування реляційними базами даних у хмарному середовищі [77]. Цей інструмент використовується для оптимізації управління та розгортанням баз даних, що базуються на моделях відносин між даними. AWS RDS надає високий рівень автоматизації, дозволяючи користувачам фокусуватися на розробці та аналізі даних, мінімізуючи при цьому необхідність управління апаратною інфраструктурою. Цей сервіс підтримує різні типи реляційних баз даних, включаючи MySQL, PostgreSQL, Oracle, Microsoft SQL Server та Amazon Aurora. AWS RDS гарантує високу доступність і надійність даних, забезпечуючи автоматичне резервне копіювання та відновлення даних. Вибрано тип бази даних PostgreSQL, бо він підтримує можливість зберігання вкладених об'єктів та масивів.

Розглянемо схему бази даних для роботи з даними по проходженню тестів людини-оператором на Рис. 4.1. Таблиця «test» буде використовуватися для зберігання інформації потрібної для генерації самого тесту для комп'ютерного тренажеру. Кількість записів у таблиці буде обмежена кількістю спроектованих тестів. Дані до таблиці будуть додаватися адміністратором, а при роботі з користувачами через програму будуть лише вичитуватися. Таблиця буде містити такі поля:

- test\_id – тип даних integer, первинний ключ таблиці.
- planes – тип даних jsonb, обмеження – не може містити значення null. Зберігає масив об'єктів з даними про розташування літаків на карті для кожного тесту, кожен об'єкт містить пару географічних даних про розташування на карті latitude і longitude, тип позначки літака та розмір позначки літака на карті.
- visibility – тип даних double precision, обмеження – не може містити значення null. Поле відповідає за відсоток видимості об'єкта на карті.

- `start_position` – тип даних `jsonb`, обмеження – не може містити значення `null`.  
Об'єкт, що містить інформацію про початкове розташування карти при початку роботи над тестом.

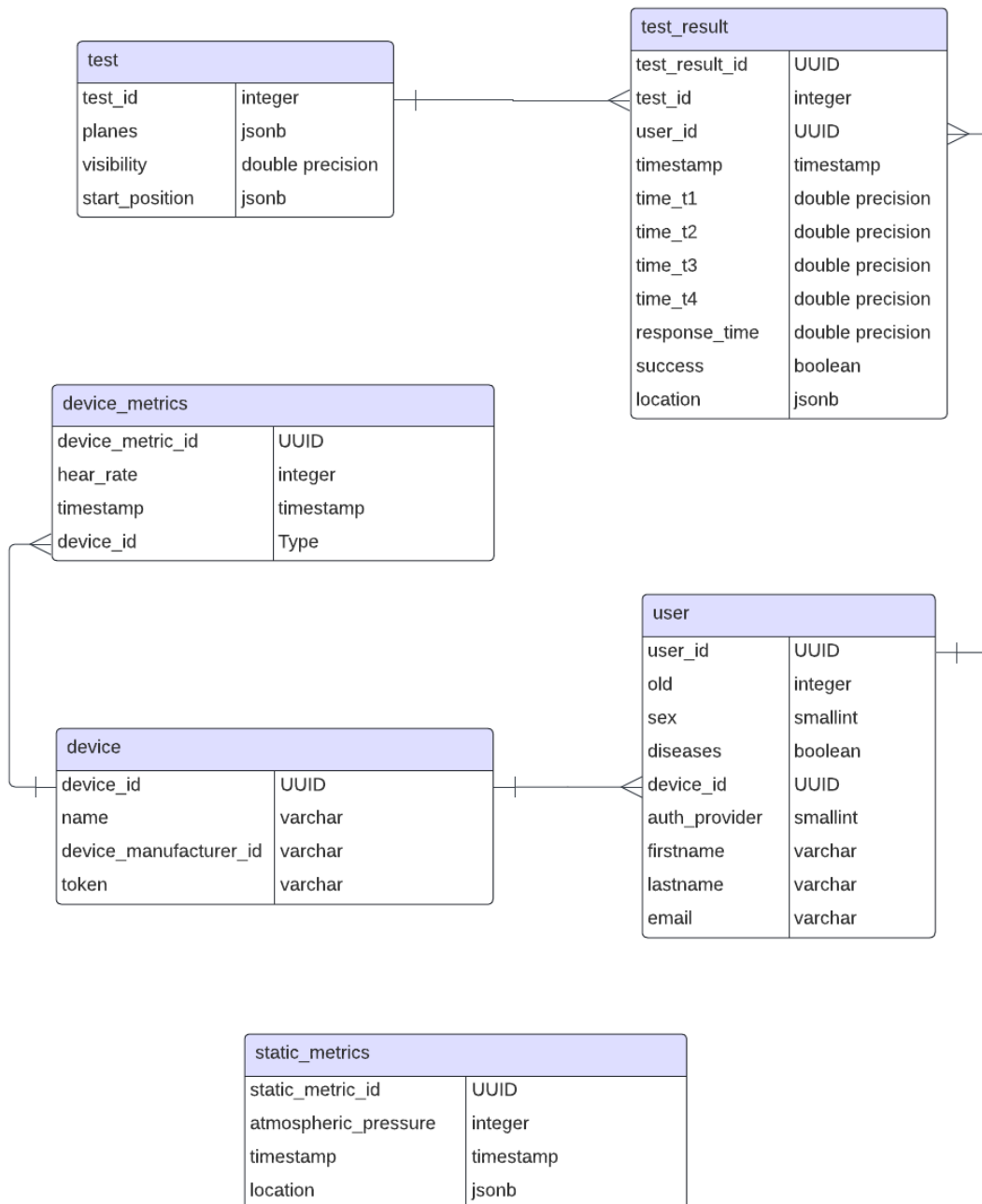


Рис. 4.1. Схема реляційної бази даних «тестування операторів за допомогою комп'ютерного тренажера»

Таблиця «`test_result`» матиме ціль для зберігання інформації про виконання тесту людиною-оператором. Під час роботи оператор буде додавати дані до таблиці по кожному окремому тесту. Також ця таблиця містить основні дані, що

будуть використовуватись для аналізу параметрів оператора та впливу на успішність виконання тесту. Таблиця міститиме наступні поля:

- `test_result_id` – тип даних `UUID`, первинний ключ таблиці, генерується `postgresql` при створенні запису.
- `test_id` – тип даних `integer`, обмеження – не може містити значення `null`, зовнішній ключ, який зв’язує з таблицею «test».
- `user_id` – тип даних `UUID`, обмеження – не може містити значення `null`, зовнішній ключ, який зв’язує з таблицею «user».
- `timestamp` – тип даних `timestamp`, обмеження – не може містити значення `null`. Поле зберігає дату коли було почато роботу на тестом.
- `time_t1` – тип даних `double precision`. Поле зберігає часу у секундах витрачений на ознайомлення з завданням.
- `time_t2` – тип даних `double precision`. Поле зберігає часу у секундах витрачений на пошук об’єкта.
- `time_t3` – тип даних `double precision`. Поле зберігає часу у секундах витрачений на аналіз чи правильний об’єкт вибрав користувач під час тесту.
- `time_t4` – тип даних `double precision`. Поле зберігає часу у секундах витрачений на прийняття рішення по завершенню завдання тесту.
- `response_time` – тип даних `double precision`, обмеження – не може містити значення `null`. Поле зберігає весь час у секундах витрачений на тест.
- `success` – тип даних `boolean`, обмеження – не може містити значення `null`.
- `location` – тип даних `jsonb`. Поле зберігає об’єкт, що містить дані про геолокацію.

Таблиця «user» матиме ціль зберігати дані про користувача, що проходить тестування та буде мати наступні поля:

- `user_id` – тип даних `UUID`, первинний ключ таблиці, генерується `postgresql` при створенні запису.
- `old` – тип даних `integer`, обмеження – не може містити значення `null`. Поле зберігає вік користувача.

- `sex` – тип даних `smallint`. Зберігає значення про стать користувача.
- `diseases` – тип даних `boolean`. Поле відобрадає чи є хронічні хвороби у користувача.
- `device_id` – тип даних `UUID`, обмеження – не може містити значення `null`, зовнішній ключ, який зв’язує з таблицею «`device`».
- `auth_provider` – тип даних `varchar(100)`. Зберігає дані про провайдера для автентифікації.
- `firstname` – тип даних `varchar(100)`. Поле містить ім’я користувача.
- `lastname` – тип даних `varchar(100)`. Поле містить прізвище користувача.
- `email` – тип даних `varchar(100)`. Зберігає електронну пошту користувача.

Таблиця «`device`» буде використовуватись для зберігання даних про самі пристрої з сенсорами для опрацювання даних типу пульсу та міститиме наступні поля:

- `device_id` – тип даних `UUID`, первинний ключ таблиці, генерується `postgresql` при створенні запису.
- `name` – тип даних `varchar(100)`, обмеження – не може містити значення `null`. Поле зберігає ім’я пристрою.
- `device_manufacturer_id` – тип даних `varchar(100)`, обмеження – не може містити значення `null`. Поле зберігає унікальний ідентифікатор для пристрою, що використовується для отримання даних по пристрою.
- `token` – тип даних `varchar(255)`, обмеження – не може містити значення `null`. Зберігає значення токена для отримання даних по пристрою.

Таблиця «`device_metrics`» міститиме параметри подані вже з самого пристрою у відрізок часу. У таблиці будуть присутні такі поля:

- `device_metric_id` – тип даних `UUID`, первинний ключ таблиці, генерується `postgresql` при створенні запису.
- `heart_rate` – тип даних `integer`, обмеження – не може містити значення `null`. Зберігає дані про пульс.
- `device_id` – тип даних `UUID`, обмеження – не може містити значення `null`, зовнішній ключ, який зв’язує з таблицею «`device`».

- `timestamp` – тип даних `timestamp`, обмеження – не може містити значення `null`. Поле зберігає дату коли було отримано параметри з пристрою.

У таблиці «`static_metrics`» буде зберігатися інформація про параметри навколишнього середовища, що впливають на роботу людини-оператора. Таблиця не пов'язана з іншими таблицями так як дані можуть бути спільними для багатьох тестів. Міститиме наступні поля:

- `static_metric_id` – тип даних `UUID`, первинний ключ таблиці, генерується `postgresql` при створенні запису.
- `atmospheric_pressure` – тип даних `integer`, обмеження – не може містити значення `null`. Зберігає дані про атмосферний тиск у міліметрах ртутного стовпа.
- `timestamp` – тип даних `timestamp`, обмеження – не може містити значення `null`. Поле зберігає дату коли відібрано параметри середовища.
- `location` – тип даних `jsonb`. Поле зберігає об'єкт, що містить дані про геолокацію.

## 4.2. Архітектура системи

Наступним кроком після проектування сховища даних – є розділення системи на компоненти та проектування контрактів взаємодії між ними. Першочергово йде проектування підсистеми комп'ютерного тренажеру. Діаграма прецедентів для даної підсистеми зображено на Рис. 4.2. Маємо двох акторів: адміністратор та людина-оператор. Людина оператор має такі прецеденти:

- переглядати завдання;
- виконувати завдання;
- переглядати власні результати тестування.

Адміністратор має ціль наповнювати систему та перевіряти результати, тому має наступні прецеденти:

- додавати завдання;
- переглядати завдання;
- переглядати результати всіх користувачів;
- завантажувати файли з даними тестування.

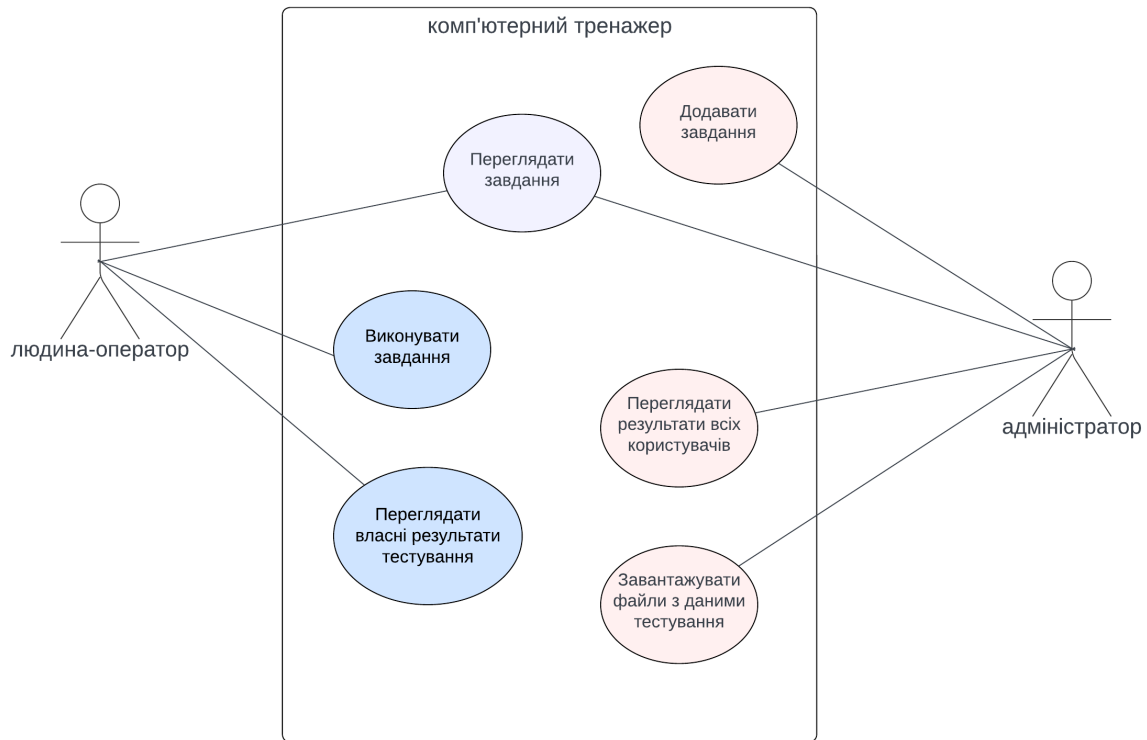


Рис. 4.2. Діаграма прецедентів

Дана підсистема буде мати клієнт-серверну архітектуру взаємодії через протокол TCP/HTTP. Під час такої взаємодії потрібно врахувати важливість автентифікації та авторизації користувачів. Роутинг та моніторинг також потрібні для даної системи. Враховуючи ці потреби, має сенс використати API Gateway, що візьме на себе ці обов'язки.

Завдання іншої підсистеми буде використання методів класифікації, кластеризації, навчання та показ статистики. Ця частина буде використовувати бібліотеки для роботи з машинним навчанням, статистикою, класифікацією та іншими. Для успішної роботи такої системи потрібно мати дані, тобто потрібно створити API для експорту даних, які згенерували оператори комп'ютерного тренажеру. Ці дані будуть містити значення параметрів. Варто додати два варіанти API для експорту даних без значень null чи пустих значень та з усіма даними. Це дасть змогу оцінити адміністратору наскільки чисті дані відповідають сирим даним та спробувати зробити самостійну чистку для оптимізації.



Розгортання системи є важливим кроком. Дана система передбачає не постійну роботу, особливо на початкових етапах. Архітектура «безсерверна» (Serverless) в Amazon Web Services (AWS) є підходом до розробки та виконання додатків, де видача ресурсів серверами та управління ними покладається на хмарний провайдер, такий як AWS [78]. Безсерверна архітектура передбачає не лише уникнення простою в роботі самого додатку, а й баз даних та інших ресурсів. Основні переваги, які пропонує AWS для даної архітектури:

- Споживання ресурсів за замовчуванням. У безсерверній архітектурній моделі ви сплачуєте тільки за те, що використовуєте. Це означає, що вам не потрібно оплачувати за простій чи неактивність. Ресурси (включаючи введення/виведення) обчислюються на основі кількості фактично використовуваних ресурсів та тривалості їх використання.
- Швидкість розгортання. Розгортання серверних застосунків відбувається швидше, оскільки розробники можуть концентруватися на написанні коду та функціональності, а не на управлінні інфраструктурою. AWS автоматично розмірковує та розподіляє ресурси.
- Автоматизоване масштабування. Система автоматично масштабується відповідно до навантаження. Наприклад, при великому обсязі запитів або збільшенні завантаження, AWS може автоматично масштабувати кількість екземплярів, щоб забезпечити стабільну продуктивність [79].
- Низький рівень адміністрування і технічної складності. Безсерверні сервіси в AWS позбавлені багатьох завдань технічного адміністрування, таких як управління серверами, конфігурацією мережі та обробка патчів. Це робить архітектуру менш технічно складною для розробників.
- Інтеграція з іншими AWS сервісами. AWS Serverless інтегрується з іншими хмарними сервісами AWS, такими як Amazon S3 для зберігання, Amazon RDS для баз даних, а також з іншими сервісами для моніторингу та журналювання. AWS Cognito для менеджменту користувачів та їх доступу.

Хоча архітектура безсерверна не підходить для всіх випадків використання, вона є привабливою для багатьох додатків, особливо тих, які мають непередбачувані або варіюючі обсяги трафіку, що відповідає вимогам системи.

На Рис. 4.3 продемонстровано компоненти системи з використанням безсерверної архітектури, використовуючи хмарні сервіси від Amazon. Діаграма показує взаємодію між різними компонентами системи. Клієнтом виступає веб-додаток, який також розгорнутий в середині AWS Cloud. Людина-оператор, або ж просто користувач, виконує певну роботу на сайті і для цього клієнт взаємодіє з API Gateway, щоб клієнт взаємодіяв з відповідними API на сервері. Для цього серверний додаток працює в AWS App Runner. AWS App Runner дає можливість запускати веб-сервери, написані на більшості сучасних мов програмування. Але перед тим як взаємодіяти з веб-сервером клієнт має пройти автентифікацію та авторизацію для доступу до певного ресурсу [80]. Після цього відбувається комунікація за допомогою HTTP запитів між клієнтом та сервером. Сервер опрацьовуючи запити клієнта виконує запити до бази даних. Amazon RDS підтримує усі стандартні інтерфейси взаємодії такі як JDBC чи ODBC драйвери.

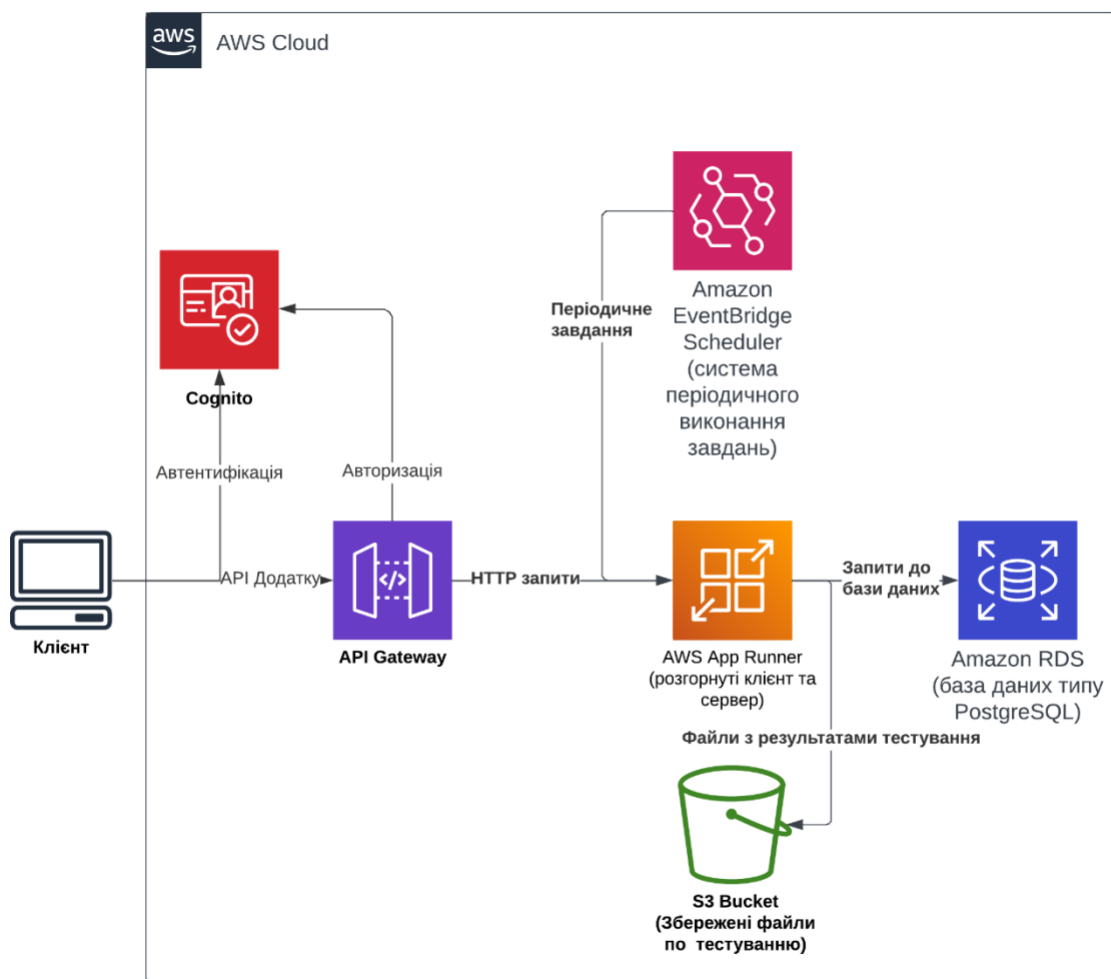


Рис. 4.3. Діаграма безсерверної архітектури системи

Також на діаграмі присутній Amazon EventBridge Scheduler. Це сервіс, що виконує запуск певних завдань за розкладом [81]. З метою зберігання даних у csv файлах по кожному окремому дню та зріз усіх даних вкінці кожного дня є така система запуску завдань за розкладом. Технічно вона запускає певний процес у самому веб-сервері. Після цього дані з бази даних RDS завантажуються у вигляді csv файлів до сервісу зберігання файлів Amazon S3. Далі файли використовуються адміністратором для роботи з методами класифікації та кластеризації.

Наступним кроком розглянемо діаграму діяльності роботи користувача в системі на Рис. 4.4. На діаграмі бачимо, що система генерує послідовність для користувача.

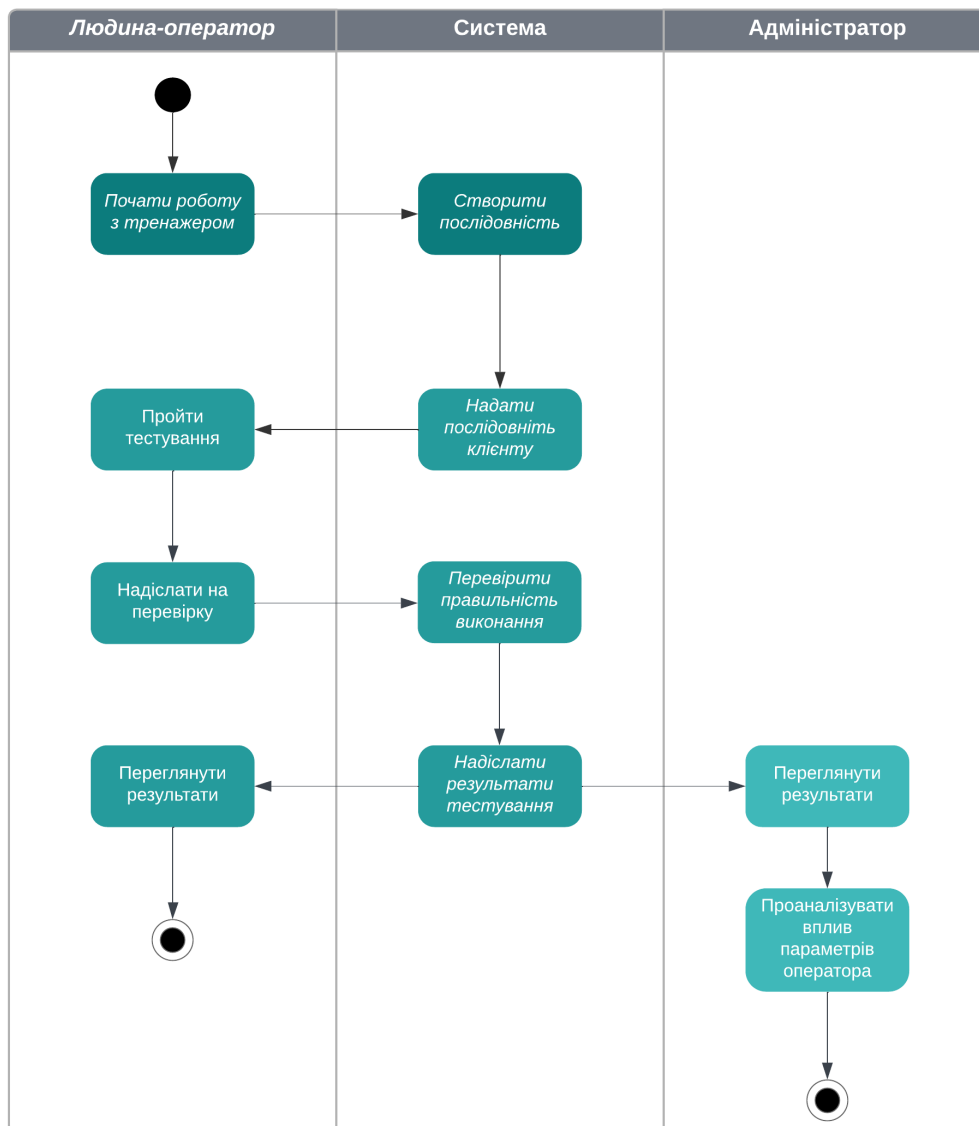


Рис. 4.4. Діаграма діяльності роботи користувача в системі

Послідовність потрібна для того, щоб користувач міг проходити тестування кілька разів, але в різних порядках. Це зроблено з ціллю уникнення запам'ятовування кроків один за одним при повторному тестуванні. Даліше оператор надсилає результати на перевірку, які перевіряє система. Користувач отримує результати лише вкінці тестування. Коли система завершує перевірку то результати отримує людина-оператор та адміністратор.

Адміністратор переглядає результати та може виконати аналіз нових даних, вплив різних параметрів оператора на його результати, знаходження аномалій.

Даліше розглянемо детальніше в яких станах може перебувати оператор під час тестування. Під час виконання тесту користувач проходить кілька етапів:

- Ознайомлення з завданням. Користувач читає текст завдання, у випадку пошуку літака він дивиться номер літака та його тип, а також що потрібно зробити з цим літаком.
- Пошук об'єкту. Виконує пошук об'єкту на карті. Даний крок виділяє час пошук відповідного типу літака, а саме його значка на карті.
- Аналіз об'єкту. Перевірка номеру літака об'єкта.
- Прийняття рішення. На завершальному кроці користувач приймає рішення, що потрібно зробити з літаком.

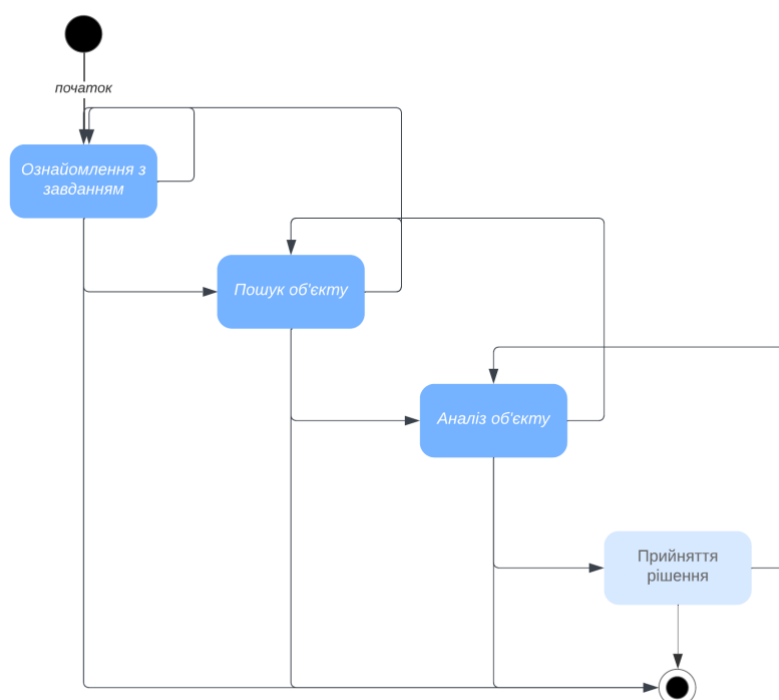


Рис. 4.5. Діаграма станів при проходженні тесту людино-оператором

На Рис. 4.5. можна побачити стани, що відповідають кожному кроку під час проходження тесту. Початковим є стан «ознайомлення з завдання» і крок за кроком оператор переходить до завершення тесту. Проте під час кожного зі станів оператор може повернутися до попереднього. Це може бути через сумнів або ж через те що початково було вибрано не вірний об'єкт, і так можна дійти аж до початкового стану, що ще раз ознайомитись з завданням. Особливістю є те що під час будь-якого стану може закінчитись час і автоматично буде завершено тест, тому з будь-якого стану можна перейти до кінця.

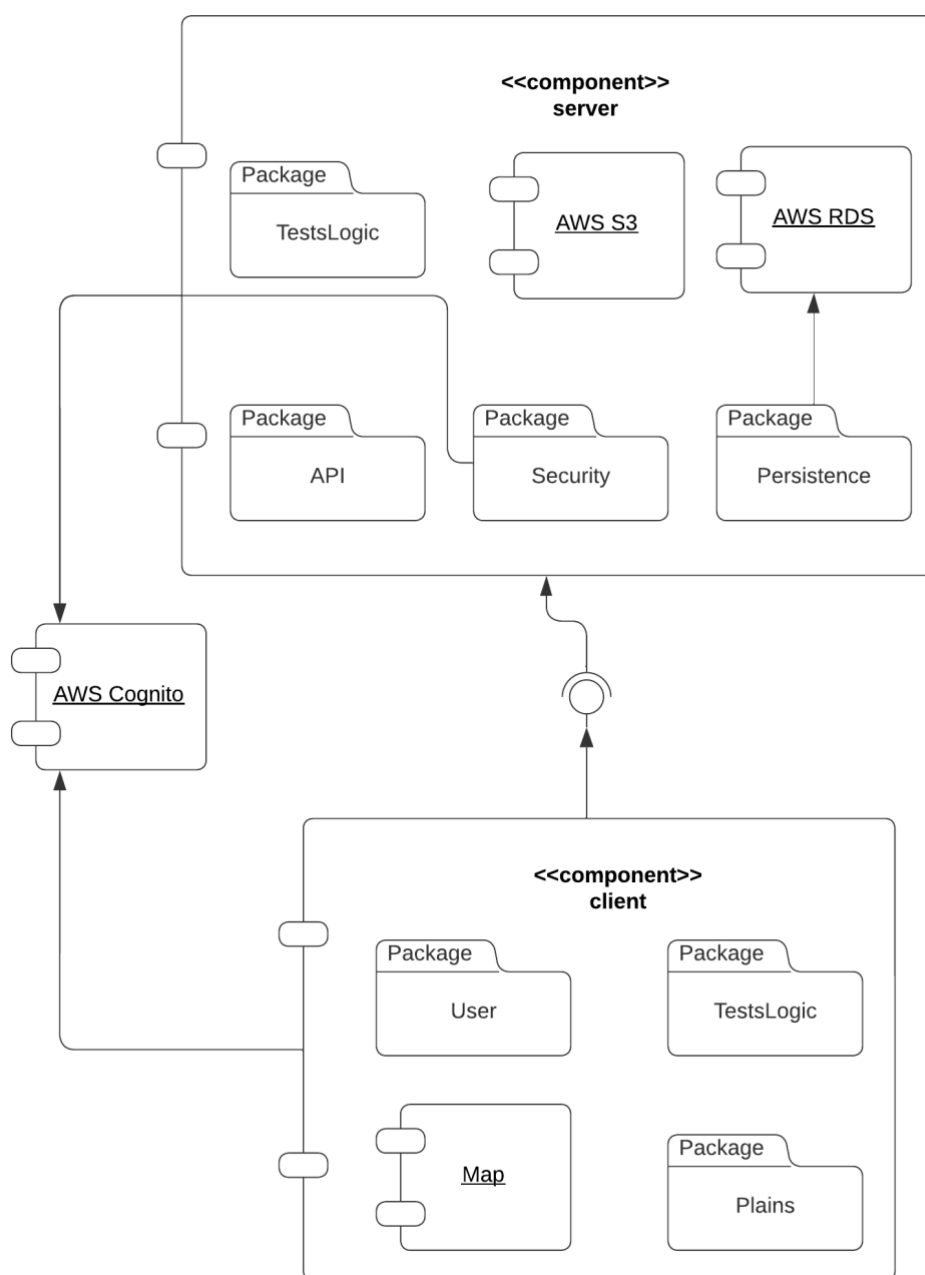


Рис. 4.6. Діаграма компонентів

Діаграма компонентів зображена на Рис. 4.6. Представлено дві основні розроблені компоненти: client і server. Client включає в себе компоненти карту та пакети: TestsLogic – логіка для роботи з тестами, User – автентифікація та авторизація користувача, Plains – відображення літаків. Компонента server включає в себе компоненти: AWS RDS – база даних, AWS S3 – хмара для файлів. Також до серверної компоненти входять пакети: Persistence – для роботи з базою даних; Security – для автентифікації та авторизації користувачів, а також взаємодія з AWS Cognito; API – набір http обробників для запитів від клієнтської частини; TestsLogic – логіка для роботи з тестами, що зберігається на сервері.

### 4.3. Розробка системи

#### 4.3.1. Розробка веб-застосунку

Для розробки веб-застосунку було обрано мову програмування TypeScript та фреймворк Angular. TypeScript – є розширенням JavaScript, мова призначена для введення статичної типізації та додаткових можливостей в розробку програмного забезпечення. Дозволяючи програмістам визначати типи змінних та параметрів функцій, TypeScript пропонує численні переваги порівняно з традиційним JavaScript. Однією з ключових переваг TypeScript є його здатність виявляти помилки на етапі компіляції, що сприяє покращенню стабільності та надійності програмного коду. Система статичної типізації дозволяє виявляти потенційні проблеми перед запуском програми, що сприяє зниженню кількості помилок під час виконання.

Додатково, TypeScript дозволяє розробникам використовувати сучасні функціональності ECMAScript та ООП, такі як класи, інтерфейси та модулі, що забезпечує більшу чіткість та структурованість коду [82]. Це робить проект розширеного обсягу більш легким для управління та розвитку, що особливо корисно в середовищі великих проектів. Узагальнюючи, TypeScript виступає як потужний інструмент у сфері розробки програмного забезпечення, дозволяючи розробникам писати більш безпечний, структурований та документований код, зменшуючи ймовірність помилок та покращуючи ефективність розробки.

Angular – це платформа та фреймворк для розробки веб-додатків, створений компанією Google. Визначається він як високопродуктивний, масштабований та ефективний інструмент для створення сучасних веб-додатків. Angular базується на мові програмування TypeScript і пропонує об'єктно-орієнтований підхід до розробки. Основні характеристики Angular включають:

- Компонентна архітектура. Angular реалізує компонентну архітектуру, де кожен веб-елемент є компонентом [83]. Компоненти є самостійними, перевикористовувані та добре організовані частинами коду, що полегшує розробку та підтримку додатків.
- Модульність. Angular сприяє модульності шляхом організації функціональності у модулі. Модулі дозволяють групувати компоненти, сервіси та інші ресурси, що полегшує організацію коду та його масштабованість.
- Двостороннє зв'язування даних. Angular надає можливість встановлення двостороннього зв'язування даних між елементами інтерфейсу користувача та логікою додатка. Це спрощує синхронізацію даних між моделлю та представленням.
- Ін'єкція залежностей. Фреймворк використовує принцип ін'єкції залежностей для організації та управління компонентами та сервісами. Це дозволяє створювати незалежні та легко тестовані компоненти.
- Розширені можливості роутингу. Angular надає потужний механізм роутингу, що дозволяє створювати односторінкові додатки з ефективним керуванням станами та навігацією.
- Компіляція та оптимізація. Angular використовує AOT (Ahead-of-Time) компіляцію для перетворення коду в JavaScript та HTML ще до виконання додатка [84]. Це підвищує продуктивність та забезпечує більш ефективну роботу додатків.

Використання мови програмування TypeScript та технології Angular дають можливість розробити сучасний веб-сайт та не потребують глибокого вивчення технологій як React. Сучасні технології веб розробки не є предметом дослідження

дисертаційної роботи тому має сенс використати найпростішу з сучасних технологій для представлення результатів.

Сам веб-застосунок також використовує карту для відображення літаків та інших об'єктів. Використана типова карта, що призначена для відслідковування літаків у реальному часі. На самій карті позначаються об'єкти різного розміру з якими працює оператор. Важливим кроком в клієнтській частині є також відслідковування часу по різних діях користувача. Ці всі дані беруться з активності маніпуляторів таких як комп'ютерна миша, трекпад чи клавіатура.

### 4.3.2. Розробка веб-сервера

Для розробки серверної частини системи, тобто API модуля, використовується мова програмування Java та фреймворк Spring Boot. Java, як високорівнева, об'єктно-орієнтована мова програмування, вирізняється рядом характеристик, які визначають її важливість і широке застосування в сучасному програмуванні. Дана мова програмування має вже довгу історію та суттєвий розвиток від свого першого релізу. Зі стабільних загальновідомих переваг Java можна виділити платформонезалежність, безпеку, масштабованість, широка екосистема та велика спільнота розробників. Останні релізи принесли не тільки вдосконалені можливості по ООП розробці, а також багато патернів та механізмів зі світу функціонального програмування. Розглянемо деякі нові механізми Java:

- Optional монада – надає можливість безпечно працювати з об'єктами, що можуть мати значення null [85].
- Функціональні інтерфейси надають синтаксичний цукор та заставляють дотримуватися принципу єдиною відповідальності [86].
- Можливість ділити систему на модулі та використовувати лише потрібні модулі як залежність, що дозволяє зменшити розмір застосунку.
- Ключове слово var, завдяки якому можна оголошувати змінні не вказуючи тип, але тип буде присутній так як це статична мова програмування.

Для проведення тестування потрібна певна послідовність тестів, тому для проведення тестування користувач має кілька варіантів послідовності. Це може



бути звичайний порядок по номеру тесту, випадковий чи зворотній. Щоб зберігати послідовність тестів використовується колекція `ConcurrentHashMap`.

Фреймворк `Spring Boot` – це продовження самого `Spring`. `Spring Boot` приніс сильно спрощений процес налаштування та запуску застосунків, а також можливість інтегруватися з різними базами даних, чергами, системами роботи з великими даними, потоковим передаванням(`streaming`). Також є можливість моніторингу застосунку та повна підтримка усіх потрібних механізмів для розгортання та стабільної роботи(`liveness, readiness probes`) в `Kubernetes` кластері.

Для побудови системи використано `Spring Data` для роботи з базою даних `AWS RDS`. Технологія має сильні переваги та багато готових механізмів для роботи. Достатньо додати дані для підключені до бази даних: порт, мережевий адрес, пароль, ім'я користувача та назва бази даних. Наступним кроком є створити класи, що відображають таблиці в базі даних. `Object-Relational Mapping` – це концепція та підходи до розробки програмного забезпечення, які спрощують взаємодію між об'єктно-орієнтованою моделлю даних (наприклад, класами у мові програмування) та реляційною базою даних [87].

Використано мову програмування `Java` та технологію `Spring/Spring Boot` для розробки серверної частини веб-сайту з метою мати можливість розгортати застосунок на будь-яких платформах, також `Java` підтримує повноцінну інтеграцію зі `Spark`, що може бути корисним при подальшому розвитку. Також є підтримка усіх необхідних технологій, що дозволяє зекономити час при розробці аплікації.

Для автентифікації користувачів є можливість використовувати анонімний доступ, коли користувач вказує лише свій вік. Також якщо користувач має бажання переглядати результати свого тестування у майбутньому, а не лише зразу після тесту то він може зареєструватися. Коли система веде облік користувачів то важливим кроком є надійність та безпека. Менеджмент паролів з одного боку є досить простим завданням та існує багато відомих механізмів та патернів по роботі з даними типу паролів користувачів, але все ж це додатковий час та інколи вимагає підтримки коли знаходяться безпекові прогалини в певних технологіях. Тому в даній системі використано `AWS Cognito`. `Amazon Cognito` – це сервіс

ідентифікації та управління доступом, наданий Amazon Web Services (AWS). Він призначений для реалізації безпеки, автентифікації та авторизації користувачів в додатках та веб-сервісах [88]. AWS Cognito забезпечує простий і масштабований спосіб керування користувачами та їх автентифікацією без необхідності великих зусиль з вашого боку. Основні можливості AWS Cognito включають:

- Реєстрація та вхід користувачів. AWS Cognito дозволяє реєструвати та автентифікувати користувачів з використанням різних методів, таких як ім'я користувача та пароль, або використання сторонніх ідентифікаторів, таких як Google або Facebook.
- Управління доступом та авторизацією. Сервіс надає засоби для налаштування прав доступу користувачів до ресурсів вашого додатка чи сервісу. Можливості авторизації включають ролі, групи та правила авторизації.
- Синхронізація даних користувача. AWS Cognito дозволяє синхронізувати дані користувачів між різними пристроями та платформами, забезпечуючи єдиної інтерфейс для доступу до даних.
- Двофакторна автентифікація. AWS Cognito дозволяє включити додатковий шар безпеки через двофакторну автентифікацію, забезпечуючи додатковий захист для облікових записів користувачів.

Комунікація між клієнтом та сервером виконується по HTTP протоколу та підтримує принципи REST. Для автентифікації та авторизації використовується JWT токен. JWT (JSON Web Token) – це стандарт для представлення даних про користувача між сторонами в JSON-форматі [89]. JWT складається з трьох частин: заголовку (Header), твердження (Payload) та підпису (Signature). Кожна з цих частин кодується у Base64 та об'єднується за допомогою крапки. Клієнт-сервер має такі основні API взаємодії:

- `/tests/start` – тип запиту POST. Цей запит ініціює початок тестування та повертає посилання на перший тест.
- `/tests/{testId}` – тип запиту GET. «testId» – це унікальний номер тесту. У результаті цього запиту сервер повертає об'єкт у якому є номер тесту та

масив з даними про розташування літака на карті(latitude та longitude), його розмір на карті та значок як саме відображається літак.

- /tests/{testId}/test-results – тип запити POST. «testId» – це унікальний номер тесту. Цей запит містить дані про проходження тесту, а саме загальний час загальний та по кроках виконання. У відповідь клієнт отримує посилання на наступний тест, у випадку закінчення отримує 208, що тестування вже закінчено.
- /users/{userId}/test-results – тип запити GET. «userId» – це унікальний ідентифікатор користувача. Повертає масив з результатами тестування. Інформація по кожному окремому тесту та загальна по всьому тестуванню.
- /export/csv – тип запити GET. Запит доступний для адміністратора, щоб завантажити всі дані по тестуванню за весь час. Повертає csv файл.

Для навчання використовуються дані з розумного пристрою. Під час дослідження та збору даних використовувався власний розумний годинник Garmin Fenix 6X Pro. Існує два варіанти як ці дані можуть потрапити в систему: зайти на сайт виробника і завантажити файл з даними за певний період або використати API і система сама буде завантажувати дані за період коли відбувалось тестування. На перших кроках було використано варіант з ручним додаванням даних до бази даних і синхронізації по часу, але після кількох етапів тестування було використано API. Перевагою API є можливість роботи з різними виробниками розумних пристроїв: Garmin, Apple, FitBit, Polar, Withings та інші. Використовуючи API є можливість задавати проміжки часу для яких потрібні дані та які саме дані потрібні, тобто можна задавати лише час проходження тестування та пульс, щоб уникнути надмірного використання ресурсу в API.

У випадку атмосферного тиску також було використано API. Метеорологічні дані будуть спільними для кількох людей якщо вони проходять тестування в одному місці та в один час тому дані не є зв'язані з іншими сутностями. Було використано API meteo matics, який надає можливість задати локацію та час.

### 4.3.3. Розробка методів аналізу даних

Задачі аналізу даних, класифікації та кластеризації були виконанні за допомогою мови програмування Python та фреймворки Pandas і scikit-learn. Переваги використання Python для класифікації та кластеризації:

- Широкий вибір бібліотек машинного навчання. Python має потужний екосистему бібліотек машинного навчання, таких як scikit-learn, TensorFlow, PyTorch і інші. Ці бібліотеки надають реалізації різних алгоритмів класифікації та кластеризації, а також інструменти для оцінки та візуалізації результатів [90].
- Зручність для реалізації моделей машинного навчання. Python – це мова програмування загального призначення, яка дозволяє легко реалізовувати та налаштовувати моделі машинного навчання. Велика кількість прикладів коду, документації та спільнота сприяють прискоренню процесу розробки.
- Легкість інтеграції з іншими інструментами. Python добре інтегрується з іншими інструментами та бібліотеками для аналізу даних та візуалізації, такими як NumPy, SciPy, Matplotlib і Seaborn. Це дозволяє легко комбінувати різні інструменти для повного циклу аналізу даних.
- Зручність в розробці та налагодженні. Python має читабельний та лаконічний синтаксис, що робить код легко зрозумілим та зручним для розробників. Це дозволяє ефективно вирішувати завдання класифікації та кластеризації та вдосконалювати моделі

Головними завданнями до технологій, що використовуються для реалізації методів аналізу даних є можливість роботи з простими csv файлами, щоб уникнути витрачання ресурсів на підготовку даних до вичитування. Також необхідно бібліотеку для відображення гістограм, дендрограм, дерев рішень та інших типів діаграм чи графіків і це все є присутнє та працює по стандарту в бібліотеках мови Python. На етапі дослідження поки не планується для впровадження, але для перспективи також варто зазначити момент, що Python має повну підтримку з роботою з Apache Spark, звісно це не є пряма інтеграція як у Scala. Apache Spark написаний на Scala і тому всі бібліотеки є напряму

підключені з вихідного коду Spark API. Але підтримка pyspark на високому рівні і присутня вся необхідна інтеграція.

Саме навчання та застосування методів класифікації та кластеризації відбувалось в Google Colab. Google Colab – це безкоштовний сервіс від Google, який надає можливість виконувати та публікувати Jupyter-ноутбуки у хмарі [91]. Цей сервіс дозволяє вам працювати з кодом Python, створювати та редагувати ноутбуки, використовуючи ресурси Google для виконання обчислень. Colab надає обчислювальні ресурси, такі як центральний процесор (CPU) та графічний процесор (GPU), безкоштовно. Це особливо корисно для виконання завдань з машинного навчання, які вимагають великої потужності обчислень. Colab дозволяє використовувати графічні процесори (GPU) та Tensor Processing Units (TPU) для прискорення обчислень, що особливо важливо для навчання моделей глибокого навчання. Ви можете легко імпортувати та експортувати дані, збережені на Google Drive, що робить спільну роботу з проектами зручною та забезпечує зручний доступ до файлів. Загально, Google Colab став дуже популярним інструментом для аналізу даних, машинного навчання та наукових досліджень завдяки своїм безкоштовним ресурсам та легкій інтеграції з іншими інструментами Google.

#### **4.4. Апробація результату**

Розглянемо готовий веб-застосунок комп'ютерного тренажеру побудований для людей-операторів, зображений на Рис. 4.7. Кожен користувач має завдання пройти тестування на тренажері, що імітує роботу оператора авіаційного радару або карти відслідковууючи літаки, що перебувають в зоні роботи оператора.

Про проект

### Завдання

Завдання розділені по 3 рівнях складності: 10 - першого рівня, 20 - другого рівня, 10 - третього рівня.

Різниця у складності полягає в розмірі об'єктів пошуку, кількості шуму на карті, та розмитості фону.

Для початку тестування виберіть 'Пройти тестування' і ви будете перенаправлені на карту з тренажером.

Кожне завдання має обмеження по часу 30 секунд, але також буде відслідковуватись по етапах виконання завдання.

### Етапи

**t1** - час у секундах витрачений на ознайомлення з завданням

**t2** - час у секундах витрачений на пошук об'єкта

**t3** - час у секундах витрачений на аналіз чи правильний об'єкт вибрав користувач під час тесту

**t4** - час у секундах витрачений на прийняття рішення по завершенню завдання тесту

Рис. 4.7. Веб-застосунок комп'ютерного тренажеру

Користувач, людина-оператор, не знає наперед яка буде початкова позиція лише після натискання кнопки «Почати тестування» система згенерує послідовність тестування і користувач побачить карту відслідковування літаків по типу зображену на Рис. 4.8. Знайшовши об'єкт пошуку користувач переглядає його додаткову інформацію та бачить усю інформацію про літак. Приймаючи рішення людина-оператор вибирає, що потрібно зробити з літаком, подати сигнал до нього чи подати дані про нього до диспетчерської служби, для цього присутні відповідні кнопки зображені на Рис. 4.9.

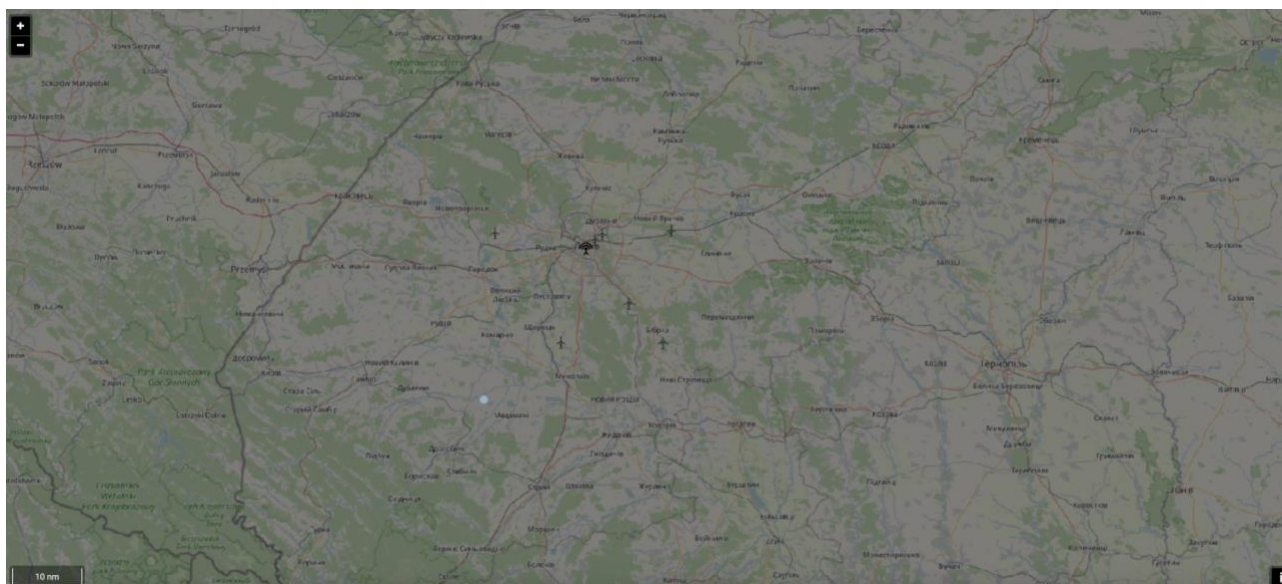


Рис. 4.8. Процес проходження тестування

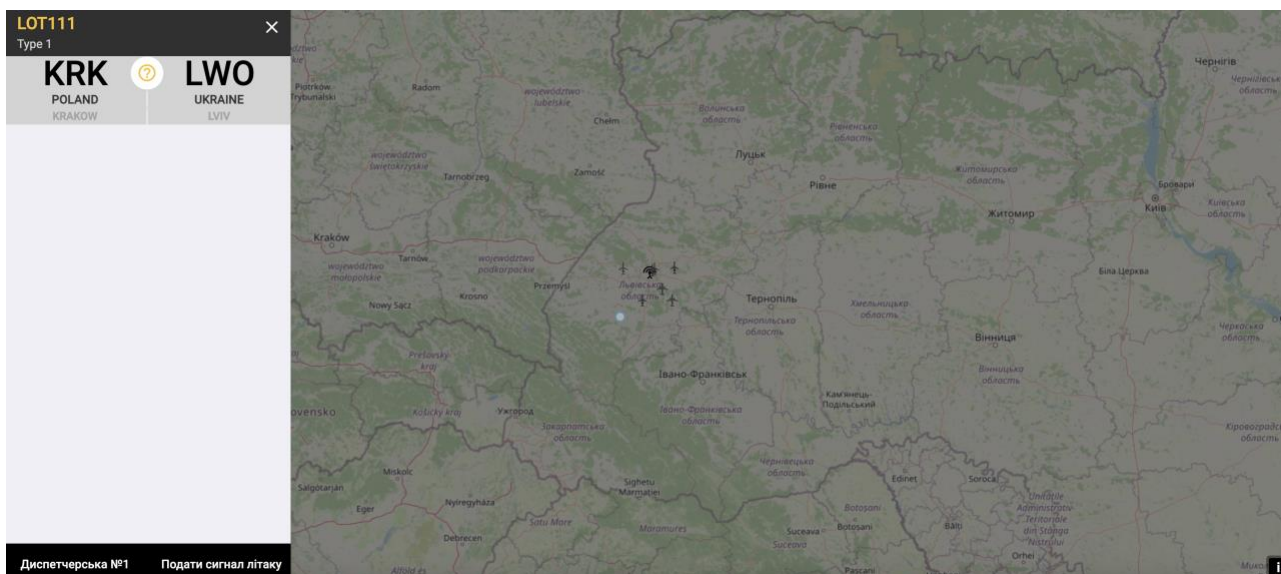


Рис. 4.9. Завершення проходження тесту

Після завершення усіх тестів користувач переходить до стадії завершення тестування. На Рис. 4.10. зображено результати тестування користувача по всіх тестах. Також якщо користувач має бажання він може зареєструватися і отримувати дані по кількох колах тестування, що можуть проведені в різні дні. Рівні відрізняються кількістю об'єктів шуму та розмиттям фону, тому важливим є показати людини-оператору його результати по рівнях.

Комп'ютерний тренажер

Про проект

Пройти тестування

Богдан Худоба

Результати тестування: 85% правильно вирішених завдань.

Завдання були поділені по 3 рівнях складності: 10 - першого рівня, 20 - другого рівня, 10 - третього рівня.

Не вірно вирішені завдання були в наступних тестах: 3, 11, 14, 22, 38, 39

Тести першого рівня

90%

Тести другого рівня

85%

Тести третього рівня

80%

Рис. 4.10. Результати отримані користувачем

Окрім статистичних даних по всіх тестуваннях система збирає дані по кожному кроку, що дає змогу користувачеві розуміти на які етапи він витрачає більше часу. Зображено приклади розподілу часу між кроками тестування на Рис. 4.11. для позитивного вирішення тесту і на Рис. 4.12. для неуспішного виконання.

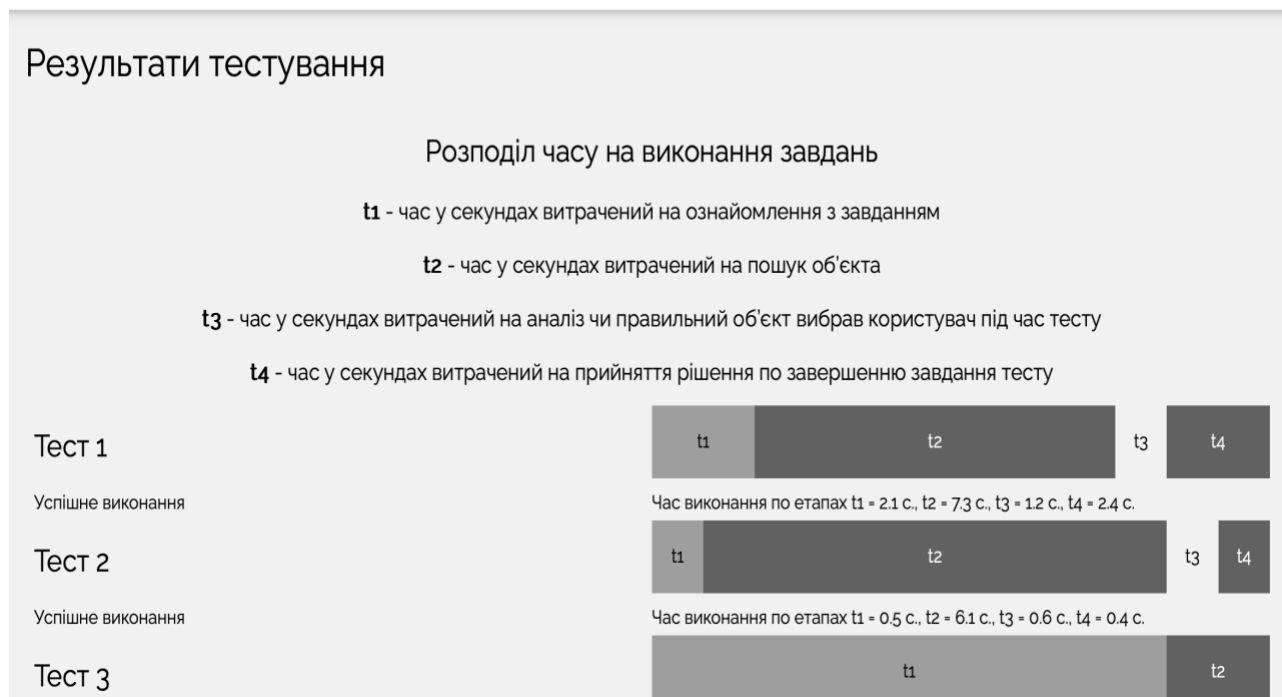


Рис. 4.11. Результати роботи успішного тесту по етапах

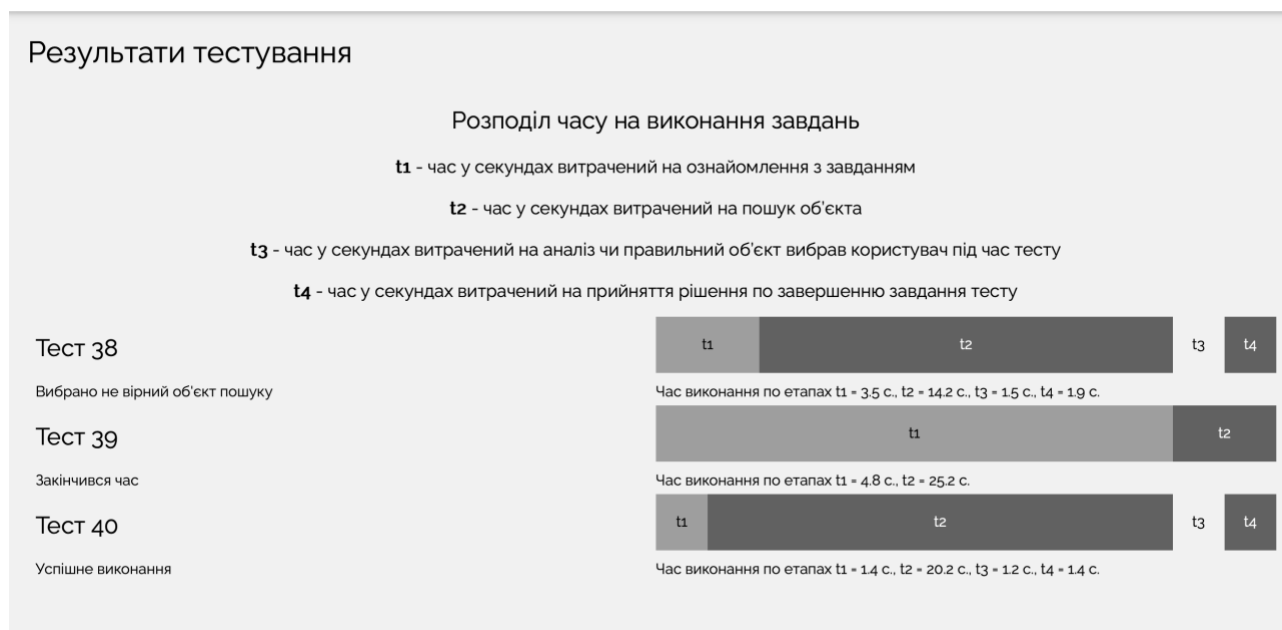


Рис. 4.12. Результати роботи неуспішного тесту по етапах



Результати по етапах показують імпульси відпочинку та прийняття рішення, що є моментами зміни концентрації уваги та стресовими моментами. Завдяки цьому людини-оператор може самостійно визначити, коли та за яких умов відбувалося найбільше стресових ситуацій.

Розглянемо перегляд результатів в режимі адміністратора зображено на Рис. 4.13. Його роль передбачає відбір персоналу, надання рекомендацій та проведення аналізу параметрів на результативність операторів.

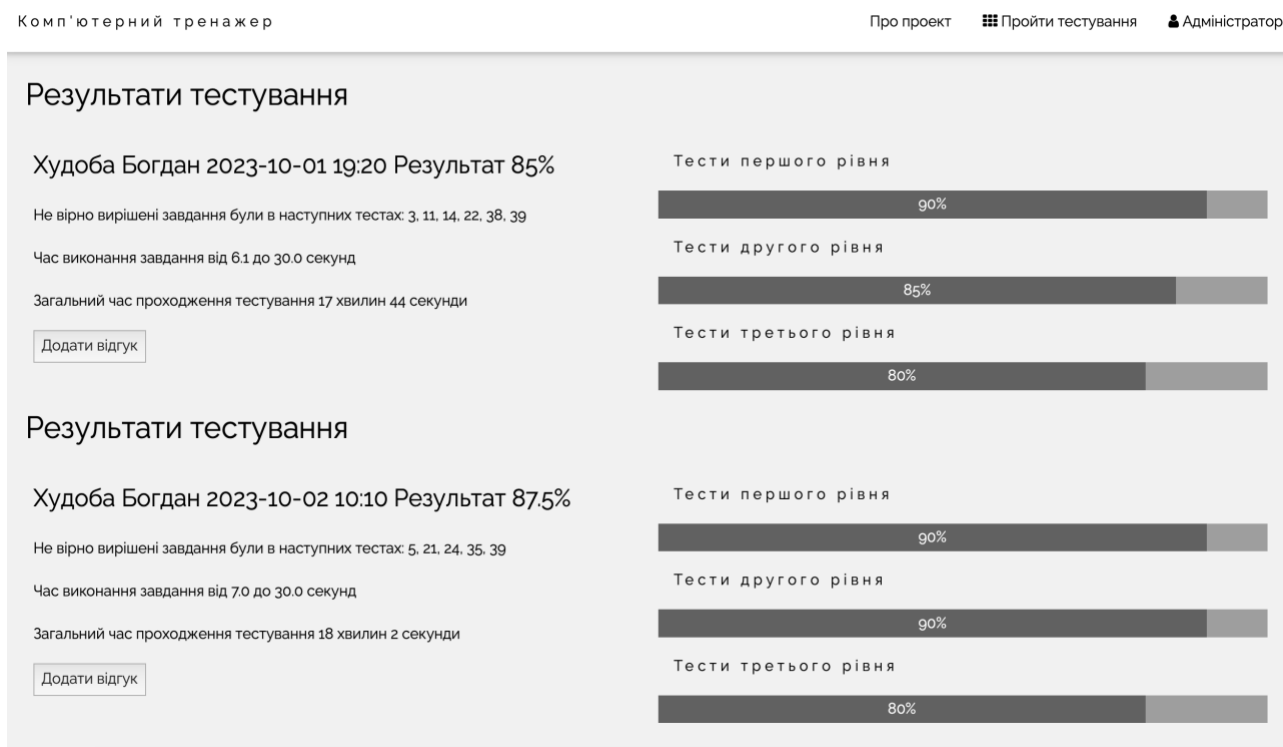


Рис. 4.13. Перегляд результатів в режимі адміністратора

Побачивши статистику по певних завданнях та по витраченому часу на різних кроках проходження тесту, адміністратор може зробити висновки про те чи були якісь чинники які негативно впливають на оператора. Також адміністратор бачить всю статистику по зміні пульсу людини-оператора. Враховуючи всі дані, що бачить адміністратор, він також може надати рекомендації для операторів, процес зображено на Рис. 4.14.

Беручи до уваги, успішність різних людей-операторів адміністратор може здійснити експрес відбір операторів для відповідної роботи. Можливість перегляду по рівнях дає можливість краще компонувати робочі зміни, коли є високий рівень навантаження чи низький. Тобто операторів з високим рівнем

стресостійкості поставити на зміну, коли присутня велика кількість рейсів у прикладі роботи авіадиспетчера.

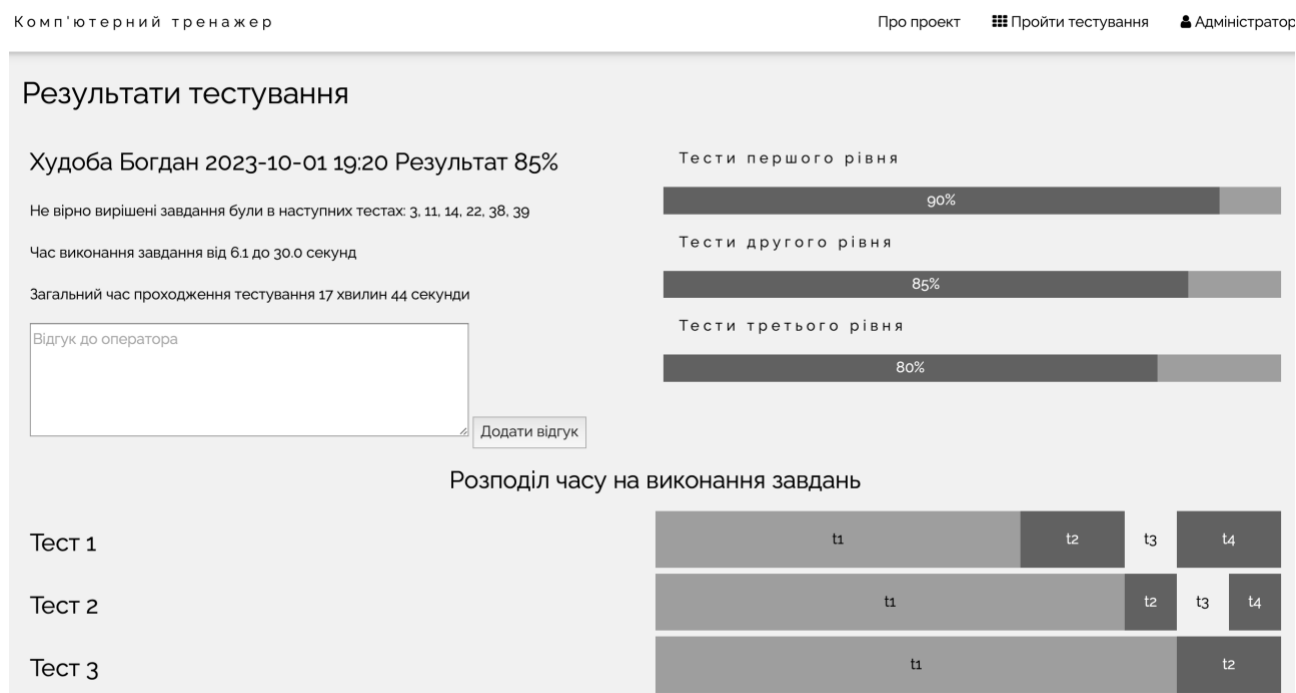


Рис. 4.14. Надання рекомендацій від адміністратора до користувача

Також при наданні рекомендацій є можливість згенерувати звіт для отримання повних результатів по проходженню тестування. Людина-оператор після отримання рекомендацій має можливість пройти тестування ще раз з метою покращення своїх результатів після отримання рекомендацій. Для людини-оператора також є можливість отримання звіту по всіх тестах. У додатках Б.1 та Б.2 показано приклад даних, що може отримати людина-оператор.

## Висновки до четвертого розділу

У четвертому розділі було спроектовано сховище даних, у якому зберігаються параметри оператора для аналізу, а також технічні дані потрібні для отримання даних з розумних пристроїв. Також у даному розділі було подано діаграму прецедентів, яка показує основні варіанти використання системи комп'ютерного тренажеру та розподіл завдань між людиною-оператором та адміністратором. Було подано діаграму безсерверної архітектури для розгортання в хмарному сервісі Amazon Web Services. Аргументовано причини використання даної

архітектури для проекрованої системи. Додано механізм для зберігання файлів необхідних для навчання та перевірки точності на нових даних, збережено файли по днях та вкінці кожного дня.

Розроблено діаграму діяльності для людини-оператора, адміністратора і самої системи. Виділено кроки взаємодії між різними сутностями. Окрім діаграми діяльності побудовано діаграму станів для проходження одного окремого тесту, що показує як людина-оператор проходить тест. Враховуючи типовість кожного тесту діаграма станів для одного тесту відображає циклічний процес роботи людини-оператора.

Аргументовано використання технологій для реалізації системи. Враховано особливості клієнтської частини – веб-застосунку, серверної частини – веб-сервера та частини побудови методів аналізу даних. Також подано опис основних АРІ для взаємодії між клієнтом та сервером.

Розроблено користувацький інтерфейс, проілюстровано його роботу за допомогою скріншотів, показано етапи проходження тестування та перегляду результатів для людини-оператора.

Проаналізувавши вимоги до проекрованої системи та отримані результати можна дійти до висновку, що основні цілі та вимоги були досягнуті при створенні системи.

Запропоновані методи і моделі впроваджені у навчальний процес Національного університету «Львівська політехніка» при викладанні дисципліни «Інтелектуальний аналіз даних» (підтверджено актом впровадження).

Також результати дисертаційної роботи впроваджено в ПП «Скіфи» (підтверджено актом впровадження).

Впровадження в ДБ «Інформаційна технологія формування психофізичного портрету в умовах стресових ситуацій» (№ держ. реєстру 0119U002257).

## ВИСНОВКИ

У дисертаційній роботі розв'язано актуальне наукове завдання розроблення методів та засобів аналізу параметрів оператора з використанням комп'ютерних тренажерів.

1. Здійснено аналіз літератури, включаючи порівняльний аналіз з сучасними аналогічними дослідженнями аналізу параметрів оператора. Це показало що дослідження використовують більше параметри систем, а не самого оператора. Розглянуто стан інформаційних технологій для моделювання стану людини-оператора.
2. Організовано комп'ютерний тренажер, що відповідає моделі людино-машинного інтерфейсу для виконання пошук об'єкту. Комп'ютерний тренажер забезпечено набором тестових завдань для симуляції пошукової діяльності. Також тренажер фіксує усі кроки, що виконує оператор та зберігає дані для подальшого аналізу. Це дало можливість отримати параметри оператора при роботі за комп'ютерним тренажером.
3. Проведено аналіз впливу складності тестових завдань на діяльність та ефективність оператора з використанням даних отриманих з експериментального дослідження. Також розроблено математичну модель людино-машинного інтерфейсу для відслідковування станів оператора під час виконання завдання пошуку об'єктів. Було використано математичну модель східцевої функції для відслідковування стану людини-оператора в різні проміжки часу. Оператор перебуває у кожному стані певний час і переходить в інший, провівши кілька етапів тестування на графіку періодичної функції видно зміни в проміжках часу перебування в різних станах. Це дає розуміння чи не було зовнішнього впливу або стресових ситуацій.
4. Організовано систему експрес оцінювання операторської діяльності. Для цього було використано показник Герста та фрактальну розмірність, що дало можливість використати нові показники для кращої оцінки операторів та їх рангування під час експрес відбору.

5. Було удосконалено метод ієрархічний класифікатор, використовуючи попередню кластеризацію та динамічне зважування, що дало змогу підвищити індекс Джіні до 0.49 у батьківському вузлі та побудувати збалансоване дерево рішень.
6. Здійснено розробку архітектуру системи з орієнтацією на веб-сайт. Аргументовано використання безсерверної архітектури для розгортання системи. Проведено аналіз та вибір технологій для реалізації людино-машинної системи. Розроблено веб-сайт для експрес оцінки параметрів операторської діяльності. Це дало змогу за допомогою простого рішення здійснювати експрес оцінку операторів та давати рекомендації по відбору операторського персоналу.

**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

- 1 Камінський Р. М. Моделювання динаміки оперативності космонавта в умовах тривалого перебування на пілотованих системах // Космічна наука і технологія. – 1998. – Т.4. – № 4. – С. 156 – 165.
- 2 Сікірда Ю. В. Моделювання системи підтримки прийняття рішень авіадиспетчера в позаштатних польотних ситуаціях: дис. ... канд. техн. наук: 05.13.06 / Ю.В. Сікірда. – К., 2004. – 184 с.
- 3 А.с. Комп'ютерна програма «Діагностика емоційного стану людини-оператора»: свідоцтво про реєстрацію авторського права на твір №43526 від 28.04.2012 р. / В. П. Харченко, Т. Ф. Шмельова, Ю. В. Сікірда, С. О. Астаф'єв.
- 4 Чернобильський Д. Ю., Щербина В.М. Особливості застосування комп'ютерних тренажерів у навчальному процесі при вивченні технічних дисциплін // Інформаційна безпека та інформаційні технології: збірник тез доповідей IV Всеукраїнської науково-практичної конференції молодих учених, студентів і курсантів, м. Львів, 27 листопада 2020 року. Львів ЛДУ БЖД, 2020, с.233-235
- 5 Şahin, C., Rokne, J., & Alhajj, R. (2019). Human behavior modeling for simulating evacuation of buildings during emergencies. *Physica A: Statistical Mechanics and its Applications*, 528, 121432.
- 6 Barak, O., & Tsodyks, M. (2023). Mathematical models of learning and what can be learned from them. *Current Opinion in Neurobiology*, 80, 102721.
- 7 Salas, E., Bowers, C. A., & Rhodenizer, L. (1998). It is not how much you have but how you use it: toward a rational use of simulation to support aviation training. *The International Journal of Aviation Psychology*, 8(3), 197-208.
- 8 Gaba, D. M. (2004). The future vision of simulation in healthcare. *Simulation in healthcare*, 1(1), 2-14.
- 9 Curtis, J. T., & Frantz, J. P. (2016). Game-based learning: The rise of the serious games. *Educause Review*, 51(2), 17.
- 10 Dykstra, J., & Paul, C. L. (2018). Cyber Operations Stress Survey (COSS): Studying fatigue, frustration, and cognitive workload in cybersecurity operations.

In 11th {USENIX} Workshop on Cyber Security Experimentation and Test ({CSET} 18).

- 11 Allsop, T., & Jones, C. (2018). Computer-Based Simulators for Team Training and Assessment in Healthcare: A Systematic Review. In *Healthcare Simulation Education* (pp. 43-57). Springer.
- 12 Anderson, C. A., & Dill, K. E. (2000). Video games and aggressive thoughts, feelings, and behavior in the laboratory and in life. *Journal of personality and social psychology*, 78(4), 772-790.
- 13 Shen, J., Wang, S., Zhang, Y., & Tracey, K. J. (2017). Conceptual framework of computer-based training in healthcare sector: A systematic review. *Journal of Educational Computing Research*, 55(7), 993-1015.
- 14 Cannon-Bowers, J. A., Salas, E., Blickensderfer, E., & Bowers, C. A. (1998). The impact of cross-training and workload on team functioning: A replication and extension of initial findings. *Human Factors*, 40(1), 92-101.
- 15 Rohrbach, H. M., Bausch, M., Bühren, A., & Vetter, S. Y. (2019). Training under the microscope: using a computer-based simulation in optical and surgical skills training. *Journal of optometry*, 12(4), 250-255.
- 16 Smith-Jentsch, K. A., Cannon-Bowers, J. A., Tannenbaum, S. I., & Salas, E. (2008). Guided team self-correction: Impacts on team processes and performance. *Small Group Research*, 39(3), 303-327.
- 17 Sawyer, B. D., & Sierhuis, M. (2013). Coordinative practices in the creation and use of a computer-based training simulator. *Human Factors*, 55(4), 742-758.
- 18 Wickens, C. D., & Hollands, J. G. (2000). *Engineering psychology and human performance*. Prentice Hall.
- 19 Endsley, M. R. (1995). Toward a theory of situation awareness in dynamic systems. *Human factors*, 37(1), 32-64.
- 20 Hancock, P. A., & Szalma, J. L. (2003). The future of training: Psychological and neuroergonomic perspectives. *Theoretical Issues in Ergonomics Science*, 4(1-2), 92-120.

- 21 Wiggins, M. W. (2009). A meta-analysis of the effectiveness of various cognitive-behavioral interventions for substance abuse and dependence. *NIDA Research Monograph*, 186, 104-129.
- 22 Feng, D., Rosopa, P. J., & Sibley, J. (2013). Task characteristics, user differences, and interruption: Implications for interruptive advertising in online media. *Computers in Human Behavior*, 29(1), 169-177.
- 23 Wickens, C. D. (2008). Multiple resources and performance prediction. *Theoretical issues in ergonomics science*, 9(2), 149-166.
- 24 H. Zhu, D. Liu, I. Bayley, R. Harrison, and F. Cuzzolin: Datamorphic Testing: A Method for Testing Intelligent Applications. In: *IEEE International Conference On Artificial Intelligence Testing (AITest)*, pp. 149-156. Newark, CA, USA (2019).
- 25 Sarter, N. B., & Woods, D. D. (1995). Pilot interaction with cockpit automation: Operational experiences with the flight management system. *The International Journal of Aviation Psychology*, 5(4), 303-321.
- 26 Parasuraman, R., Sheridan, T. B., & Wickens, C. D. (2008). Situation awareness, mental workload, and trust in automation: Viable, empirically supported cognitive engineering constructs. *Journal of cognitive engineering and decision making*, 2(2), 140-160.
- 27 Lee, J. D., & Moray, N. (1994). Trust, control strategies and allocation of function in human-machine systems. *Ergonomics*, 37(8), 1243-1260.
- 28 Endsley, M. R. (1995). Measurement of situation awareness in dynamic systems. *Human factors*, 37(1), 65-84.
- 29 Parasuraman, R., Sheridan, T. B., & Wickens, C. D. (2008). Situation awareness, mental workload, and trust in automation: Viable, empirically supported cognitive engineering constructs. *Journal of cognitive engineering and decision making*, 2(2), 140-160.
- 30 Stanton, N. A., Salmon, P. M., Walker, G. H., Baber, C., & Jenkins, D. P. (2005). *Human factors methods: A practical guide for engineering and design*. Ashgate Publishing, Ltd.
- 31 Lee, J. D. (1999). A case for a cognitive engineering. *International Journal of Human-Computer Interaction*, 11(1), 31-41.



- 32 Hollands, J. G., & Wickens, C. D. (1993). *Engineering psychology and human performance*. Prentice Hall.
- 33 Hart, S. G., & Staveland, L. E. (1988). Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. *Advances in psychology*, 52, 139-183.
- 34 Lupien, S. J., McEwen, B. S., Gunnar, M. R., & Heim, C. (2009). Effects of stress throughout the lifespan on the brain, behaviour and cognition. *Nature Reviews Neuroscience*, 10(6), 434-445.
- 35 Driskell, J. E., Salas, E., & Johnston, J. (1999). Does stress lead to a loss of team perspective? *Group Dynamics: Theory, Research, and Practice*, 3(4), 291-302.
- 36 Mizuno, K., Tanaka, M., Nozaki, S., Mizuma, H., Ataka, S., Tahara, T., ... & Muraoka, I. (2011). Changes in mood and cognitive performance during the menstrual cycle. *Psychology of Sport and Exercise*, 12(4), 534-541.
- 37 Sarter, N. B., & Woods, D. D. (1992). Nudging the operator and the design implications. In *Human factors in aviation* (pp. 161-192). Academic Press.
- 38 DeJoy, D. M. (1989). Psychosocial factors in occupational ergonomics. *Ergonomics*, 32(4), 369-385.
- 39 Shahabuddin Muhammad (2021) Modeling Operator Performance in Human-in-the-Loop Autonomous Systems, *IEEE Access*, Volume 9, с. 102715 – 102731, 10.1109/ACCESS.2021.3098060
- 40 Wan-Lin Hu, Claudio Rivetta, Erin MacDonald, David P. Chassin (2019) Optimal Operator Training Reference Models for Human-in-the-loop Systems, *HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES*, с. 3524-3531.
- 41 Знайти
- 42 Мочурад Л.І., Бойко Н.І., Яцків М.В. Моделювання стресової ситуації людини в автоматизованих системах управління технологічними процесами. / *Науковий вісник НЛТУ України*, 2020, т. 30, № 1. – С. 152 – 157.
- 43 Доброносова Ю. Особливості впливу медіатравматичного досвіду на самоактуалізацію особистості в культурі медіарозмаїття. Психосоціальна підтримка осіб з травмою війни: міжнародний досвід та українські реалії:

збірник матеріалів, доповідей Всеукраїнської науково-практичної конференції. Маріуполь: ДонДУУ, 2018. С. 22-25.

- 44 Психологія праці та її безпеки. Навч. Посібник. – Рівне: НУВГП 2013., с.154.
- 45 Губар А.О. Стрес як фактор впливу на адаптацію людини 2010 URL : [https://scienceandeducation.pdpu.edu.ua/doc/2010/3\\_2010/6.pdf](https://scienceandeducation.pdpu.edu.ua/doc/2010/3_2010/6.pdf)
- 46 Методи і засоби для тестування оператора поліграфа: монографія / С. М. Злепко, С. В. Тимчик, Р. С. Белзецький, Л. Г. Коваль. – Вінниця : ВНТУ, 2010. – 168 с.
- 47 Камінський Р., Шаховська Н., Худоба Б. Експериментальне дослідження та групування операторського персоналу пошукових систем в сенсі стресостійкості // Вісник Хмельницького національного університету. Серія: Технічні науки. – 2022. – № 5 (313). ст. 42-51
- 48 Nataliya Shakhovska, Roman Kaminskyi, Bohdan Khudoba, Vladyslav Mykhailyshyn, Ihor Helzhynskyi “A Novel Methodology Analyzing the Influence of Micro-Stresses on Human-Centric Environments” *Computation* 2023, 11(11), 224; <https://doi.org/10.3390/computation11110224>
- 49 Жлуктенко В.І., Наконечний С.І., Савіна С.С. Теорія ймовірностей і математична статистика. – К.:КНЕУ, 2001. 336 с.
- 50 Пасічник В.В., Шаховська Н.Б. Сховища даних: Навчальний посібник. - Львів: “Магнолія 2006”, 2008.- 496 С.
- 51 Alan. Agresti: *Categorical Data Analysis*. Wiley-Interscience, Nowy Jork, 2002.
- 52 N. Balakrishnan: *Handbook of the Logistic Distribution*. Marcel Dekker, Inc., 1991.
- 53 William H. Green: *Econometric Analysis*, fifth edition. Prentice Hall, 2003.
- 54 «Логістична регресія Wikiwand» [Онлайновий]. Доступний: [https://www.wikiwand.com/uk/%D0%9B%D0%BE%D0%B3%D1%96%D1%81%D1%82%D0%B8%D1%87%D0%BD%D0%B0\\_%D1%80%D0%B5%D0%B3%D1%80%D0%B5%D1%81%D1%96%D1%8F](https://www.wikiwand.com/uk/%D0%9B%D0%BE%D0%B3%D1%96%D1%81%D1%82%D0%B8%D1%87%D0%BD%D0%B0_%D1%80%D0%B5%D0%B3%D1%80%D0%B5%D1%81%D1%96%D1%8F) [Дата звернення: 4 12 2023].
- 55 «XGBoost Documentation – xgboost 2.0.2 documentation» [Онлайновий]. Доступний: <https://xgboost.readthedocs.io/en/stable/> [Дата звернення: 4 12 2023].

- 56 Sagi, Omer; Rokach, Lior (2021). "Approximating XGBoost with an interpretable decision tree". *Information Sciences*. 572 (2021): 522-542.
- 57 Yoav Freund and Robert E. Schapire (1997); A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting *Journal of Computer and System Sciences*, 55(1):119-139
- 58 Zhou Zhi-Hua (2012). *Ensemble Methods: Foundations and Algorithms*. Chapman and Hall/CRC. c. 23
- 59 Friedman, J. H. (1999). *Stochastic gradient boosting*. Stanford University.
- 60 Hastie, T., Tibshirani, R., Friedman, J. H. (2001). *The elements of statistical learning: Data mining, inference, and prediction*.
- 61 Breiman, L. (1996). Bagging Predictors. «*Machine Learning*, 24»: pp. 123–140.
- 62 Everitt, Brian (2011). *Cluster analysis*. Chichester, West Sussex, U.K: Wiley.
- 63 Bradley, Paul S.; Fayyad, Usama M. (1998). "Refining Initial Points for k-Means Clustering". *Proceedings of the Fifteenth International Conference on Machine Learning*.
- 64 Pelleg, Dan; Moore, Andrew (1999). "Accelerating exact k -means algorithms with geometric reasoning". *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*. San Diego, California, United States: ACM Press. pp. 277–281.
- 65 Forgy, Edward W. (1965). "Cluster analysis of multivariate data: efficiency versus interpretability of classifications". *Biometrics*. 21 (3): 768–769.
- 66 R. Sibson (1973). SLINK: an optimally efficient algorithm for the single-link cluster method. *The Computer Journal (British Computer Society)* 16 (1): 30–34.
- 67 Rokach, Lior, and Oded Maimon. «Clustering methods.» *Data mining and knowledge discovery handbook*. Springer US, 2005. 321—352.
- 68 B. Dubuc, J. F. Quiniou, C. Roques-Carmes, C. Tricot, and S. W. Zucker (1989). Evaluating the fractal dimension of profiles. *Phys. Rev. A*. 39: 1500–12. doi:10.1103/PhysRevA.39.1500
- 69 Олена Деза, Мішель Марі Деза. Глава 19. Відстані на дійсній та цифровій площинах. 19.1. Метрики на дійсній площині // *Енциклопедичний словник*

відстаней = Dictionary of Distances. — М. : Наука, 2008. — С. 276. — ISBN 978-5-02-036043-3.

- 70 Rokach, Lior, and Oded Maimon. «Clustering methods.» Data mining and knowledge discovery handbook. Springer US, 2005. 321—352.
- 71 Lance G.N., Willams W.T.» A general theory classification of sorting strategies. 1. Hierarchical systems // Comp. J. 1967. № 9. P. 373—380.
- 72 R. Polikar, L. Udpa, S. S. Udpa, and V. Honavar, “Learn++: An incremental learning algorithm for supervised neural networks,” IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews, vol. 31, no. 4, pp. 497–508, 2001
- 73 H. S. Mohammed, J. Leander, M. Marbach, Polikar, and R. Polikar, “Can AdaBoost.M1 learn incrementally? A comparison to Learn++ under different combination rules,” International Conference on Artificial Neural Networks (ICANN2006) in Lecture Notes in Computer Science, vol. 4131, pp. 254–263, Springer, 2006
- 74 M. D. Muhlbaier, A. Topalis, and R. Polikar, “Learn++.NC: combining ensemble of classifiers with dynamically weighted consult-and-vote for efficient incremental learning of new classes,” IEEE Transactions on Neural Networks, vol. 20, no. 1, pp. 152–168, 2009
- 75 Yang, P., Hwa Yang, Y., Zhou, B., Zomaya, Y., et al.: A review of ensemble methods in bioinformatics. Current Bioinformatics 5(4), 296–308 (2010)
- 76 Н. Б. Шаховська, Ю. Я. Болюбаш, О. М. Верес // Наукові праці Донецького національного технічного університету. Серія : Обчислювальна техніка та автоматизація. - 2014. - № 2. - С. 147-155
- 77 «What is Amazon Relational Database Service (Amazon RDS)?» [Онлайновий]. Доступний:<https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Welcome.html> [Дата звернення: 4 12 2023].
- 78 «What is serverless computing? | Serverless definition» [Онлайновий]. Доступний: <https://www.cloudflare.com/learning/serverless/what-is-serverless/> [Дата звернення: 4 12 2023].

- 79 Park, Andrew; Denlinger, Darrell; Watson, Coburn (September 18, 2015). "Creating Your Own EC2 Spot Market".
- 80 «Managed Container Apps Service – AWS App Runner - AWS» [Онлайновый]. Доступный: <https://aws.amazon.com/apprunner/> [Дата звернення: 4 12 2023].
- 81 «What is Amazon EventBridge Scheduler?» [Онлайновый]. Доступный: <https://docs.aws.amazon.com/scheduler/latest/UserGuide/what-is-scheduler.html> [Дата звернення: 4 12 2023].
- 82 «TypeScript: JavaScript With Syntax For Types.» [Онлайновый]. Доступный: <https://www.typescriptlang.org/> [Дата звернення: 4 12 2023].
- 83 «angular/angular: Deliver web apps with confidence» [Онлайновый]. Доступный: <https://github.com/angular/angular> [Дата звернення: 4 12 2023].
- 84 «Angular - Ahead-of-time (AOT) compilation?» [Онлайновый]. Доступный: <https://angular.io/guide/aot-compiler> [Дата звернення: 4 12 2023].
- 85 «Is java.util.Optional a Monad?» [Онлайновый]. Доступный: <https://medium.com/geekculture/does-java-util-optional-is-monad-492911fb66ee> [Дата звернення: 4 12 2023].
- 86 Herbert Schildt, Java The Complete Reference 9th edition, 2014
- 87 Douglas Barry, Torsten Stanienda, "Solving the Java Object Storage Problem," Computer, vol. 31, no. 11, pp. 33-40, Nov. 1998, [<https://www.computer.org/csdl/magazine/co/1998/11/ry033/13rRUxC0SRY>]. Excerpt at <https://www.service-architecture.com/articles/object-relational-mapping/transparent-persistence-vs-jdbc-call-level-interface.html> Lines of code using O/R are only a fraction of those needed for a call-level interface (1:4). For this exercise, 496 lines of code were needed using the ODMG Java Binding compared to 1,923 lines of code using JDBC.
- 88 «Authentication Service – Customer IAM - Cognito» [Онлайновый]. Доступный: <https://aws.amazon.com/cognito/> [Дата звернення: 4 12 2023].
- 89 «JSON Web Tokens – jwt.io» [Онлайновый]. Доступный: <https://jwt.io/> [Дата звернення: 4 12 2023].

- 90 «Data Science & Python» [Онлайновий]. Доступний:  
[https://www.w3schools.com/datascience/ds\\_python.asp](https://www.w3schools.com/datascience/ds_python.asp) [Дата звернення: 4 12 2023].
- 91 «Welcome to Colaboratory» Google, [Онлайновий]. Доступний:  
<https://colab.research.google.com/?hl=ua> [Дата звернення: 4 12 2023].

## ДОДАТОК А. ПРОГРАМНИЙ КОД РЕАЛІЗАЦІЇ КЛАСТЕРИЗАЦІЇ ТА ІЄРАРХІЧНОГО КЛАСИФІКАТОРА

```
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
import pandas as pd
from scipy.cluster.hierarchy import linkage, dendrogram

df_re = pd.read_csv('results.csv', low_memory=False)
df_re
# ALL
df = df_re[['index', 'old', 'P1', 'P2', 'P3', 'T1', 'T2', 'T3', 'W', 'A', 'B']]
data_scaler = StandardScaler()
scaled_data = data_scaler.fit_transform(df)
scaled_data.shape
complete_clustering = linkage(scaled_data, method="complete",
metric="euclidean")
average_clustering = linkage(scaled_data, method="average", metric="euclidean")
single_clustering = linkage(scaled_data, method="single", metric="euclidean")
dendrogram(complete_clustering)
plt.show()

# Level 1
df = df_re[['index', 'old', 'P1', 'T1', 'W', 'A', 'B']]
data_scaler = StandardScaler()
scaled_data = data_scaler.fit_transform(df)
scaled_data.shape
complete_clustering = linkage(scaled_data, method="complete",
metric="euclidean")
average_clustering = linkage(scaled_data, method="average", metric="euclidean")
single_clustering = linkage(scaled_data, method="single", metric="euclidean")
dendrogram(complete_clustering)
plt.show()

# Level 2
df = df_re[['index', 'old', 'P2', 'T2', 'W', 'A', 'B']]
data_scaler = StandardScaler()
scaled_data = data_scaler.fit_transform(df)
scaled_data.shape
complete_clustering = linkage(scaled_data, method="complete",
metric="euclidean")
average_clustering = linkage(scaled_data, method="average", metric="euclidean")
single_clustering = linkage(scaled_data, method="single", metric="euclidean")
dendrogram(complete_clustering)
plt.show()

# Level 3
df = df_re[['index', 'old', 'P3', 'T3', 'W', 'A', 'B']]
data_scaler = StandardScaler()
scaled_data = data_scaler.fit_transform(df)
scaled_data.shape
complete_clustering = linkage(scaled_data, method="complete",
metric="euclidean")
average_clustering = linkage(scaled_data, method="average", metric="euclidean")
single_clustering = linkage(scaled_data, method="single", metric="euclidean")
dendrogram(complete_clustering)
plt.show()
```

**ДОДАТОК Б. ТАБЛИЦІ РЕЗУЛЬТАТІВ ТЕСТУВАННЯ ОПЕРАТОРІВ**

Таблиця Б.1. Часові результати проходження тестів

Номер оператора	Мін, с	Q1, с	Медіана, с	Q3, с	Макс, с
1	5.49	15.6	21.03	24.51	29.79
2	5.77	16.16	21.47	25.07	29.74
3	6.26	16.81	20.69	23.24	29.17
4	5.7	15.79	19.76	24.54	29.88
5	5.91	16.02	20.72	23.93	39.91
6	5.17	14.01	20.14	24.11	29.48
7	5.32	15.3	20.39	23.69	29.71
8	5.25	16.86	21.38	23.73	29.55
9	5.09	14.55	19.45	23.69	29.04
10	5.17	15.04	19.82	24.87	29.56
11	5.52	14.95	19.01	22.5	30.0
12	5.89	14.62	20.23	23.93	28.95
13	5.18	14.11	19.04	23.37	29.47
14	6.26	14.35	19.2	23.02	29.15
15	5.22	15.56	20.56	23.86	30.0
16	6.01	14.61	18.66	23.79	29.74
17	6.06	16.32	20.75	24.26	29.12
18	5.96	15.93	21.32	24.64	29.92
19	5.48	15.85	20.89	23.82	29.45
20	5.66	14.84	20.24	23.13	29.59
21	5.08	16.16	21.22	24.47	29.81
22	6.98	15.71	20.03	23.73	29.68
23	5.73	15.69	20.49	25.01	30.0
24	5.53	14.53	20.19	24.47	29.39
25	5.93	16.02	20.8	23.9	29.47



26	5.47	14.67	20.33	23.99	29.61
27	5.3	15.21	19.45	24.0	29.49
28	5.64	16.33	21.15	24.12	30.0
29	7.33	17.11	20.71	24.66	29.77
30	5.65	17.01	21.38	24.13	29.55
31	6.88	16.44	20.37	24.4	28.45
32	5.79	16.02	20.49	23.57	29.82
33	5.13	15.32	18.94	23.11	30.0
34	5.87	14.75	20.02	23.48	30.0
35	5.04	13.84	20.18	23.54	29.55
36	5.04	14.36	19.63	23.59	29.53
37	5.34	14.14	20.0	24.46	29.8
38	5.11	15.23	20.59	23.92	29.85
39	5.14	13.53	18.84	23.18	30.0
40	5.67	14.23	19.37	23.95	29.45
41	5.82	16.16	21.25	24.34	29.05
42	5.84	15.31	19.84	23.96	29.69
43	6.34	14.63	19.7	24.66	30.0
44	5.83	14.31	19.91	23.69	29.0
45	5.02	14.92	19.6	23.67	29.47
46	5.01	15.13	20.16	24.17	29.82
47	5.41	14.45	19.7	23.53	29.16
48	5.14	14.96	19.5	23.91	29.4
49	7.18	15.76	20.87	24.33	29.5
50	5.04	15.02	19.5	24.53	29.35

Таблиця Б.2. Результати проходження тестів.

Номер оператора	Вік	Середній час усіх тестів, с	Середній час по правильних тестах, с	Середній час по помилкових тестах, с	Кількість правильних тестів	Кількість помилкових тестів
1	23	19.619	19.415	20.097	84	36
2	25	20.256	20.525	19.516	88	32
3	27	19.793	19.511	20.639	90	30
4	33	19.495	19.499	19.484	90	30
5	34	19.712	19.634	19.926	88	32
6	40	18.802	19.091	17.852	92	28
7	44	19.46	19.369	19.682	85	35
8	38	19.674	19.762	19.396	91	29
9	42	18.888	19.055	18.513	83	37
10	40	19.15	19.952	17.814	75	45
11	40	18.61	18.984	17.77	83	37
12	43	19.171	19.976	17.218	85	35
13	45	18.711	18.371	19.417	81	39
14	43	18.65	19.226	16.997	89	31
15	41	19.314	19.371	19.198	81	39
16	40	18.599	18.958	17.419	92	28
17	50	19.981	19.504	21.291	88	32
18	34	19.947	19.27	21.407	82	38
19	31	19.943	19.432	21.622	92	28
20	42	18.914	18.552	20.0	90	30
21	42	20.106	19.876	20.896	93	27
22	33	19.12	19.04	19.316	85	35
23	40	19.807	19.119	21.235	81	39
24	40	19.166	18.91	19.788	85	35
25	52	19.563	20.057	17.939	92	28

26	50	18.844	18.673	19.295	87	33
27	54	18.794	19.102	18.014	86	34
28	47	20.029	19.887	20.335	82	38
29	49	20.622	20.809	20.087	89	31
30	46	20.148	20.033	20.508	91	29
31	42	19.749	19.013	21.336	82	38
32	40	19.517	19.665	19.029	92	28
33	41	18.694	18.952	17.953	89	31
34	48	19.052	18.211	21.268	87	33
35	38	18.828	18.397	19.692	80	40
36	39	18.591	18.907	17.722	88	32
37	52	18.793	18.664	19.072	82	38
38	44	19.422	19.396	19.485	85	35
39	47	18.096	18.595	16.664	89	31
40	43	18.804	18.259	19.854	79	41
41	42	19.849	19.979	19.424	92	28
42	47	19.052	19.416	18.268	82	38
43	51	19.249	19.76	17.489	93	27
44	51	18.86	19.211	17.932	87	33
45	36	18.845	19.084	18.096	91	29
46	48	19.308	19.875	17.871	86	34
47	49	18.901	18.376	20.797	94	26
48	55	19.081	18.979	19.468	95	25
49	56	19.89	19.363	21.278	87	33
50	63	19.353	18.851	20.525	84	36

## ДОДАТОК В. АКТИ ВПРОВАДЖЕННЯ



ЗАТВЕРДЖУЮ

Директор ПП «Скіфі»

«08» грудня 2023 р.

### АКТ

**про впровадження результатів дисертаційної роботи  
аспіранта кафедри «Системи штучного інтелекту»  
Національного університету «Львівська політехніка»  
Худоби Богдана Петровича**

Цей акт підтверджує, що результати дисертаційної роботи Худоби Б.П. були використані для аналізу параметрів оператора у м. Кам'янка-Бузька у 2023р.

Впровадження дисертаційних досліджень Б.П. Худоби полягає у наступному:

- Розроблено комп'ютерний симулятор і генератор тестових зображень. Вони дозволяють оцінити стресостійкість на основі порівняння мікростресу та усвідомленості. Генерується послідовність зображень в окремі моменти. На екрані монітора оператору виставляється послідовність тестових зображень з об'єктами уваги заданого класу, і оператор повинен реалізувати відповідне рішення. Моменти їх впливу та рішення, прийняті оператором, фіксуються, а їх значення вносяться до протоколу дослідження.
  - Розроблено модель класифікації та кластеризації параметрів оператора із попереднім застосуванням аналізу аномалій. Удосконалено ієрархічний класифікатор на основі попередньої кластеризації та динамічного зважування для визначення рівня стресостійкості. Це дає змогу підвищити індекс Джіні до 0.324 в батьківському вузлі та побудувати збалансоване дерево.
- Даний акт не є підставою для взаємних фінансових розрахунків.

Директор



Дунець Олег Ярославович



про впровадження в навчальний процес результатів  
дисертаційної роботи

**Худоба Богдан Петрович**

Цей акт складено про те, що результати дисертаційної роботи Худоби Богдана Петровича впроваджено у навчальний процес кафедри "Системи штучного інтелекту" Національного університету "Львівська політехніка".

Впровадження результатів дисертаційної роботи полягає в їхньому використанні при викладанні навчальних дисциплін як окремих розділів лекційних курсів, так і в циклах лабораторних робіт.

Зокрема для викладання дисципліни «Інтелектуальний аналіз даних» для студентів освітньо-кваліфікаційного рівня «бакалавр», що навчаються за напрямом 122 "Комп'ютерні науки" використано такі результати:

- Дослідження методів класифікації та кластеризації;
- Застосування методів у комплексі.

У лекційному курсі «Основи систем штучного інтелекту» для студентів кваліфікаційного рівня «бакалавр», що навчаються за напрямом 122 "Комп'ютерні науки", використано такі результати:

- Метод-ансамбль ієрархічний класифікатор;
- Методи аугментації даних.

Директор ІКНІ,  
д.т.н., професор

 **Микола МЕДИКОВСЬКИЙ**

Завідувач кафедри СШ,  
д.т.н., професор

 **Наталія ШАХОВСЬКА**

Професор кафедри СШ,  
д.т.н., доцент

 **Наталія МЕЛЬНИКОВА**



ПІДТВЕРДЖУЮ

Проректор з наукової роботи  
 Національного університету  
 «Київська політехніка»

Іван ДЕМИДОВ

2023 р.

**використання наукових результатів  
 дисертаційної роботи Худоби Богдана Петровича,  
 представленої на здобуття наукового ступеня доктора філософії**

Комісія в складі: голови комісії - начальника науково-дослідної частини д.т.н., с.п.с Небесного Р.В. та членів комісії - завідувача кафедри СШ Шаховської Н.Б., професора кафедри СШ Яковини В.С., доцента кафедри СШ Хавалка В.М., доцента кафедри СШ Кривенчука Ю.П. цим актом підтверджують, що результати дисертаційної роботи Худоби Б.П. зокрема

- модель використання східцевих функцій для аналізу станів у часових рядах;
- метод-ансамбль класифікації результативності операторів;
- модель класифікації та кластеризації операторів

використано у науково-дослідних роботах фінансованих Міністерством освіти і науки України, що виконувались на кафедрі систем штучного інтелекту і включено до звіту: «Інформаційна технологія формування психофізичного портрету в умовах стресових ситуацій» (№ держ. реєстру 0119U002257).

Отримані автором результати використано:

- при розробленні систем аналізу параметрів об'єкта;
- при розробленні засобів збору метрик у часових рядах;
- при розробленні засобів класифікації операторів за параметрами.

Голова комісії:

начальник  
 науково-дослідної частини  
 д.т.н., с.п.с

Роман НЕБЕСНИЙ

Члени комісії:

завідувача кафедри СШ  
 професор кафедри СШ

Наталія ШАХОВСЬКА  
 Віталія ЯКОВИНА

доцент кафедри СШ

Віктор ХАВАЛКО

доцент кафедри СШ

Юрій КРИВЕНЧУК