

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Метод автоматизованої перевірки сертифікаційних авіаційних документів за допомогою штучного інтелекту»

на здобуття освітнього ступеня магістра
зі спеціальності 121 Інженерія програмного забезпечення
освітньо-професійної програми «Інженерія програмного забезпечення»

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

_____ Максим КРИСЕВИЧ
(підпис)

Виконав: здобувач вищої освіти групи ПДМ-63
Максим КРИСЕВИЧ

Керівник: _____ Тимур ДОВЖЕНКО
канд. техн. наук

Рецензент: _____ Ім'я, ПРІЗВИЩЕ
науковий ступінь,
вчене звання

Київ 2026

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**
Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти Магістр

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

_____ Ірина ЗАМРІЙ

«_____» _____ 2025 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Крисевичу Максиму Анатолійовичу

1. Тема кваліфікаційної роботи: «Метод автоматизованої перевірки сертифікаційних авіаційних документів за допомогою штучного інтелекту»

керівник кваліфікаційної роботи Тимур ДОВЖЕНКО, канд. техн. наук

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «30» жовтня 2025 р. № 467.

2. Строк подання кваліфікаційної роботи «19» грудня 2025 р.

3. Вихідні дані до кваліфікаційної роботи: науково-технічна література, методи математичне моделювання та методи машинного навчання, методи системного аналізу та проектування, методи статистичного аналізу

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Дослідження підходів та алгоритмів, які застосовуються для автоматизованої перевірки документів.

2. Дослідження специфіку сертифікаційних авіаційних документів та визначення ключові елементи, що підлягають верифікації

3. Розробка та валідація методу перевірки сертифікаційних авіаційних документів з урахуванням їх структурних особливостей

5. Перелік ілюстративного матеріалу: *презентація*

1. Методи та моделі автоматизованої перевірки документів
2. Математична модель методу перевірки сертифікаційних авіаційних документів
3. Алгоритм методу перевірки сертифікаційних авіаційних документів
4. Структура автоматизованої системи перевірки документів
5. Система оцінювання ефективності пропонуваного методу.
6. Результати моделювання методу перевірки сертифікаційних авіаційних документів.

6. Дата видачі завдання «31» жовтня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз наявної науково-технічної літератури щодо автоматизованої перевірки документів	31.10 – 05.11.2025	
2	Дослідження методів штучного інтелекту для перевірки документів	06.11 – 09.11.2025	
3	Дослідження специфіки сертифікаційних авіаційних документів	10.11 – 13.11.2025	
4	Розробка гібридного методу перевірки сертифікаційних авіаційних документів	14.11 – 16.11.2025	
5	Розробка програмної системи автоматизованої перевірки документів	17.11 – 21.11.2025	
6	Реалізація розробленого методу та проведення моделювання	22.11 – 24.11.2025	
7	Оформлення роботи: вступ, висновки, реферат	25.11 – 28.11.2025	
8	Розробка демонстраційних матеріалів	29.12 – 08.12.2025	
9	Попередній захист роботи	15.12 – 19.12.2025	

Здобувач вищої освіти

_____ (підпис)

Максим КРИСЕВИЧ

Керівник

кваліфікаційної роботи

_____ (підпис)

Тимур ДОВЖЕНКО

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 84 стор., 9 табл., 16 рис., 61 джерело.

Мета роботи – підвищення якості та ефективності перевірки сертифікаційних авіаційних документів за рахунок використання моделей та методів штучного інтелекту.

Об'єкт дослідження – процес автоматизованої перевірки сертифікаційних авіаційних документів.

Предмет дослідження – моделі та методи штучного інтелекту, що застосовуються в процесі автоматизації перевірки сертифікаційних авіаційних документів.

У роботі використано методи штучного інтелекту, графові нейронні мережі та архітектура GraphSAGE, гібридні підходи до верифікації, що поєднують машинне навчання з логічними правилами Rule-based logic.

Проведено аналіз сучасних методів автоматизованої обробки документів для класифікації та вилучення даних, виявлено їх обмеження щодо перевірки мультимодальних сертифікаційних авіаційних документів.

Розроблено гібридний метод перевірки сертифікаційних авіаційних документів, який базується на побудові мультимодального графу документа, та спроектовано архітектуру автоматизованої системи «AviaCertVerification» для реалізації цього методу.

Проведено експерименти для оцінки ефективності запропонованого методу, визначення точності класифікації та валідації документів, а також для порівняння отриманих результатів з класичними підходами, такими як TF-IDF – LR та CNN.

КЛЮЧОВІ СЛОВА: АВІАЦІЙНИЙ ДОКУМЕНТ, СЕРТИФІКАТ, МАКЕТ ДОКУМЕНТУ, ШТУЧНИЙ ІНТЕЛЕКТ, НЕЙРОННІ МЕРЕЖІ

ABSTRACT

Text part of the master's qualification work: 84 pages, 16 pictures, 9 table, 61 sources.

The purpose of the work is to improve the quality and efficiency of aviation certification document verification through the use of artificial intelligence models and methods.

Object of research – the process of automated verification of aviation certification documents.

Subject of research – artificial intelligence models and methods applied in the process of automating the verification of aviation certification documents

Summary of the work: The work employs artificial intelligence methods, graph neural networks and the GraphSAGE architecture, as well as hybrid verification approaches that combine machine learning with rule-based logic. An analysis of modern automated document processing methods for classification and data extraction was conducted, identifying their limitations regarding the verification of multimodal aviation certification documents. A hybrid method for verifying aviation certification documents was developed, based on constructing a multimodal document graph, and the architecture of the automated system "AviaCertVerification" was designed to implement this method. Experiments were conducted to evaluate the effectiveness of the proposed method, determine the accuracy of document classification and validation, and compare the obtained results with classical approaches such as TF-IDF – LR and CNN.

KEYWORDS: AVIATION DOCUMENT, CERTIFICATE, DOCUMENT LAYOUT, ARTIFICIAL INTELLIGENCE, NEURAL NETWORKS

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ	10
ВСТУП.....	12
1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ	15
1.1 Огляд та аналіз літературних джерел за темою наукового дослідження .17	
1.2 Аналіз існуючих методів, моделей та алгоритмів.....	22
1.3 Аналіз сучасних засобів програмної інженерії для автоматизованої перевірки документів	41
2 РОЗРОБКА МЕТОДУ АВТОМАТИЗОВАНОЇ ПЕРЕВІРКИ СЕРТИФІКАЦІЙНИХ АВІАЦІЙНИХ ДОКУМЕНТІВ	46
2.1 Аналіз сертифікаційної авіаційної документації України	46
2.1.1 Класифікація сертифікаційної авіаційної документації	47
2.1.2 Основні структурні елементи сертифікаційної авіаційної документації.....	50
2.1.3 Ключові ознаки сертифікаційних авіаційних документів	53
2.2 Метод перевірки сертифікаційних авіаційних документів.....	57
2.2.1 Математична модель задачі перевірки сертифікаційних авіаційних документів.....	58
2.2.2 Класифікація сертифікаційних авіаційних документів.....	62
2.2.3 Структурне моделювання документу з використанням графових нейронних мереж	63
2.2.4 Опис методу перевірки сертифікаційних авіаційних документів	64
2.3 Автоматизована система перевірки авіаційних документів	74
2.4 Програмна реалізація методу перевірки сертифікаційних авіаційних документів.....	78
2.4.1 Структура програмного проекту	78
2.4.2 Інтерфейс користувача.....	82
3 ПРОВЕДЕННЯ МОДЕЛЮВАННЯ ТА ОЦІНКА РЕЗУЛЬТАТІВ	86
3.1 Визначення метрик для оцінки ефективності розробленого методу.	86

3.2 Проведення моделювання та оцінка результатів.....	88
ВИСНОВКИ.....	93
ПЕРЕЛІК ПОСИЛАНЬ.....	95
ДОДАТОК А ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (презентація)	102
ДОДАТОК Б ЛИСТИНГИ ОСНОВНИХ МОДУЛІВ	109

ПЕРЕЛІК СКОРОЧЕНЬ

AE	- Autoencoders
AI	- Artificial intelligence -
AOC	- Air Operator Certificate
ARC	- Airworthiness Review Certificate, Authorised Release Certificate
BERT	- Bidirectional Encoder Representations from Transformers
CNN	- Convolutional Neural Network -
CofA	- Certificate of Airworthiness
CRF	- Conditional Random Fields
CRF	- Conditional Random Field
CRNN	- Convolutional Recurrent Neural Networks
CRNN	- Convolutional Recurrent Neural Network
CRS	- Certificate of Release to Service
DAE	- Denoising Autoencoders
DL	- Deep Learning
DOA	- Design Organisation Approval
EASA	- European Union Aviation Safety Agency
ELA	- Error Level Analysis
FAA	- Federal Aviation Administration
GAN	- Generative Adversarial Networks
GAN	- Generative Adversarial Networks
GNN	- Graph neural networks
GRU	- Gated Recurrent Units
HMM	- Hidden Markov Models
IDP	- Intelligent Document Processing
kNN	- k-Nearest Neighbors
LLMs	- Large Language Model
LSTM	- Long Short-Term Memory
MOA	- Maintenance Organisation Approval
NER	- Named Entity Recognition
NLP	- Natural Language Processing
OCR	- Optical Character Recognition
PDF	- Portable Document Format
POA	- Production Organisation Approval
R-CNN	- Region-based Convolutional Neural Network
RCofA	- Restricted Certificate of Airworthiness
RNN	- Recurrent Neural Networks

RoBERTa	- Robustly optimized BERT approach
RPA	- Robotic Process Automation
SIFT	- Scale-Invariant Feature Transform
SNN	- Siamese Neural Network
SNN	- Siamese Neural Network
SSD	- Single Shot MultiBox Detector
SVM	- Support Vector Machines
TC	- Type Certificate
TCDS	- Type Certificate Data Sheet
VAE	- Variational Autoencoders
ViT	- Vision Transformer
YOLO	- You Only Look Once
САП	- Спільний авіаційний простір
ШІ	- Штучний інтелект

ВСТУП

Сертифікаційні авіаційні документи уявляють собою юридичну та технічну основу глобальної системи авіаційної безпеки. Наприклад, для того, щоб літак, який було виготовлено у Сполучених Штатах Америки, міг здійснювати рейси до країн Європи та проходити сервісне обслуговування в країнах Азії, усі учасники процесу, від конструювання та виробництва літака й до виконання ним рейсів та технічного обслуговування, мають дотримуватися єдиних міжнародних стандартів. З цієї точки зору, сертифікаційний документ виступає юридичним підтвердженням від повноважного органу Державіаслужби в Україні про те, що продукт, організація або фахівець, який бере участь у цьому процесі, відповідає міжнародним стандартам.

Автоматизована перевірка сертифікаційних авіаційних документів є складною проблемою, що виходить за межі традиційних поширених підходів використання штучного інтелекту при роботі з документами. У той час, коли для обробки довільного та неструктурованого тексту ключову роль відіграють моделі обробки природної мови (NLP), у тому числі Трансформери (Transformers), а для зображень згорткові нейронні мережі (CNN), авіаційні сертифікаційні документи не можуть бути однозначно перевірені віднесенням лише до однієї з наведених груп. Такі документи уявляють собою напівструктуровані мультимодальні об'єкти, де семантичне значення окремих елементів документу може визначатися не тільки їх змістом (самі дані, як-от текст, числа, дати, зображення), а й їх просторовим компонуванням (розташуванням елементів документу на сторінці та їх відносним розташуванням).

Актуальність дослідження. Авіація є світовою індустрією. Специфіка авіаційної галузі характеризується високими вимогами до безпеки, суворою регуляцією та великими обсягами стандартизованих, але складних документів, що створює ідеальні умови для впровадження систем на базі штучного інтелекту.

Таким чином, автоматизація перевірки сертифікаційних авіаційних документів за допомогою штучного інтелекту (АІ) є умовою для підвищення безпеки, ефективності та конкурентоспроможності авіаційної галузі України.

Мета роботи: підвищення якості та ефективності перевірки сертифікаційних авіаційних документів за рахунок використання моделей та методів штучного інтелекту.

Об'єкт дослідження: процес автоматизованої перевірки сертифікаційних авіаційних документів.

Предмет дослідження: моделі та методи штучного інтелекту, що застосовуються в процесі автоматизації перевірки сертифікаційних авіаційних документів.

Для досягнення поставленої мети необхідно виконати наступні завдання.

1. Провести аналіз існуючих методів, підходів та алгоритмів, що застосовуються для автоматизованої перевірки документів.

2. Провести аналіз сучасного стану сертифікаційної документації у авіаційній галузі України.

3. Дослідити специфіку сертифікаційних авіаційних документів, їх структурні особливості та визначити ключові елементи, що підлягають верифікації автентичності таких документів.

4. Розробити та теоретично обґрунтувати метод перевірки сертифікаційних авіаційних документів, що враховує їх структурні особливості.

5. Провести аналіз та визначити метрики для оцінки ефективності розробленого методу.

6. Розробити концептуальну архітектуру автоматизованої системи перевірки авіаційних документів на основі клієнт-серверної моделі, здатну інтегрувати запропоновані методи ШІ.

7. Розробити програмне забезпечення, що реалізує розроблений метод.

8. Провести тестування та експериментальне дослідження для оцінки ефективності розробленого методу перевірки сертифікаційних авіаційних документів на основі розробленого програмного забезпечення.

Робота пройшла апробацію на науково-практичних конференціях.

1. VI Міжнародна науково-технічна конференція «Сучасний стан та перспективи розвитку IoT», 15 квітня 2025р., Київ, Державний університет інформаційно-комунікаційних технологій.

2. VI Всеукраїнська науково-технічна конференція «Застосування програмного забезпечення в ІКТ», 24 квітня 2025р., Київ, Державний університет інформаційно-комунікаційних технологій.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

Актуальність автоматизованої перевірки авіаційних документів зумовлена низкою факторів.

1. Авіаційна галузь використовує великий обсяг документації, яка включає в себе льотну, технічну документацію, документи на персонал, вантажну документацію, а також документацію адміністративно-регуляторного характеру.

2. Для авіаційної галузі критично важлива висока точність перевірки документації. Помилки можуть спричинити серйозні наслідки: поставити під загрозу безпеку польотів, призвести до затримок, штрафів та втрати репутації.

3. Авіаційні документи характеризуються складною структурою, різноманітням форматів та іноді наявністю неструктурованих даних, що значно ускладнює процес їх ручної перевірки.

4. Для забезпечення своєчасного виконання рейсів, оперативного прийняття рішень та реагування на нештатні ситуації перевірка документів повинна проводитися швидко.

5. Автоматизація процесів перевірки дозволяє значно зменшити витрати на ресурси персоналу та часу, які використовуються для перевірки документів.

6. Автоматизована перевірка документації дозволяє підвищити ефективність процесу аудиту та спростити процес підготовки звітності

Впровадження процесу автоматизованої перевірки документів має ряд переваг перед звичайною ручною перевіркою, серед яких: підвищення ефективності та якості перевірки, зниження ризику помилок, оптимізація процесу управління документацією, скорочення фінансових витрат на зберігання, витрат на оплату ручної праці та інші.

З точки зору авіаційної документації, незважаючи на супутні фінансові витрати, впровадження автоматизованої перевірки документів має такі суттєві переваги:

1) підвищення безпеки польотів за рахунок:

- підвищення точності перевірки, що знижує ризик неправильної інформації, яка може призвести до небезпечних ситуацій;
- виявлення підробок, що підвищує безпеку та запобігає несанкціонованій діяльності;
- відповідність нормативним вимогам, що є критично важливим для безпеки польотів та відповідності законодавчим вимогам до польотів;

2) підвищення ефективності обробки документів за рахунок:

- скорочення часу на обробку, що дозволяє швидше приймати рішення та уникати затримок;
- оптимізація бізнес-процесів, що дозволяє персоналу підвищити його продуктивність;
- покращення управління даними, що дозволяє використовувати централізоване зберігання, легкий доступ до документів, їх аудит, пошук необхідної інформації та формування звітності;

3) зниження фінансових затрат за рахунок:

- економії на оплаті праці персоналу (штат співробітників, зайнятих на перевірці документації може бути значно скорочений);
- зниження витрат на зберігання та обробку фізичної документації (друк, копіювання, зберігання та транспортування);
- уникнення штрафів та санкцій;

4) підвищення рівня задоволеності клієнтів за рахунок:

- прискорення реєстрації та посадки в аеропортах, у разі автоматизації перевірки документів пасажирів для цивільної авіації;
- оперативного опрацювання вантажів та перевірки вантажної документації, у разі автоматизації перевірки документів для вантажної авіації;

5) підвищення масштабованості та інтеграції з існуючими системами авіаційних перевезень та регуляторних органів.

Загальний процес автоматизованої перевірки документів включає наступні

кроки:

- 1) захоплення та оцифрування документа (за потреби, у разі, якщо документ паперовий);
- 2) попередня обробка зображення (за потреби, у разі низької якості та часткової відсутності невеликих фрагментів зображення);
- 3) оптичне розпізнавання тексту;
- 4) аналіз макета документу;
- 5) вилучення структурованої та неструктурованої інформації;
- 6) перевірка вилученої інформації на відповідність форматам, правилам, зовнішнім джерелам;
- 7) виявлення аномалій та потенційних ознак шахрайства;
- 8) перевірка автентичності візуальних елементів;
- 9) зіставлення інформації з різних частин документа або з іншими документами;
- 10) винесення рішення щодо дійсності документа та надання звіту про перевірку.

1.1 Огляд та аналіз літературних джерел за темою наукового дослідження

У контексті автоматизованої перевірки документів усе більшого значення набуває сучасний бізнес-орієнтований підхід для автоматизації повного життєвого циклу роботи з документами IDP, що автоматизує роботу з різними типами документів, використовуючи технології штучного інтелекту, машинного навчання та обробки природної мови. Технологічним ядром та набором методів для обробки цю технологію забезпечує Document Artificial Intelligence (Document AI) – галузь штучного інтелекту для обробки документів. У статті [1] автор визначає сферу Document AI, обговорює основні методи та демонструє, як ключові завдання в цій галузі можуть бути реалізовані на практиці. Основними ключовими завданнями автор визначає: оптичне розпізнавання символів OCR, аналіз макета, класифікацію

документа та витяг ключової інформації. Автор [2] детально проаналізував обробку неструктурованих документів з використанням AI, що дозволяє значно підвищити ефективність роботи з великою кількістю неоднорідних даних. Проблематика обробки багатомовних документів розкрита в [3], де запропоновано використання у поєднанні з багатомовним OCR для розпізнавання тексту з чеків. У роботі [4] розглядається інтеграція сучасних великих мовних моделей LLMs в IDP-процеси, що відкриває нові можливості для покращення точності обробки тексту. Праці [5, 6] присвячені комплексним системам управління документами, автор [5] розглядає інтегрований підхід, який поєднує методи обробки природної мови NLP, машинного навчання ML та комп'ютерного зору для автоматизації класифікації, витягування інформації та зберігання документів, автор [6] досліджує роботизовану автоматизацію процесів RPA, її взаємодію з AI та визначає вирішальну роль RPA в оптимізації обробки документів та зниженні операційних витрат за рахунок роботизованого виконання рутинних задач з документами.

Технології комп'ютерного зору є ключовим компонентом інтелектуальної обробки документів, оскільки вони забезпечують можливість «бачити» та інтерпретувати вміст документів у візуальній формі. У роботах [7,8] розглянуто еволюцію методів комп'ютерного зору: від традиційних алгоритмів до сучасних підходів, заснованих на глибокому навчанні. Зокрема, у [7] представлено критичний огляд сучасних технік глибокого навчання у комп'ютерному зорі, акцентуючи увагу на сценаріях їх застосування, включаючи аналіз документів, розпізнавання візуальних шаблонів і контекстуальну інтерпретацію зображень. Робота [8] порівнює можливості традиційних методів комп'ютерного зору з методами глибокого навчання, демонструючи переваги останніх у точності та гнучкості при вирішенні складних задач, таких як класифікація документів, виявлення елементів структури сторінки та екстракція інформації. Подальші дослідження [9,10] зосереджені на конкретних алгоритмах і підходах, які застосовуються у задачах комп'ютерного зору. У дослідженні [9] автор пропонує використання синтетичних даних для алгоритму тренування неглибокої мережі CNN+LSTM, що підвищує якість розпізнавання до 97% та використання

архітектури CNN+Transformer для розпізнавання на реальних даних, що надає до 92% точності розпізнавання. Робота [10] досліджує застосування глибоких згорткових нейронних мереж та генеративних змагальних мереж для візуального сприйняття та аналізу документів, демонструючи потенціал цих моделей у покращенні якості попередньої обробки зображень та зниженні похибок при подальшому текстовому аналізі.

Після оцифрування документа або отримання його оцифрованої версії, інтелектуальна перевірка виконується поетапно, з використанням спеціалізованих методів на кожному кроці. Серед ключових етапів – попередня обробка зображень, що забезпечує виділення важливих візуальних елементів, а також класифікація документів, яка дозволяє обрати відповідні сценарії подальшої обробки. У дослідженні [11] розглядаються сучасні досягнення у сфері застосування штучного інтелекту для обробки зображень. Описано техніки, що включають попередню обробку: підвищення якості зображень, фільтрацію шумів, а також методи глибокого навчання для виявлення об'єктів. Окрема увага приділена оптимізації моделей, що дозволяє зменшити витрати обчислень при збереженні високої точності результатів. Розглянуто приклади практичного застосування в автоматизованих системах обробки візуальних даних.

Класифікація документів, їх автоматичний розподіл по відповідним категоріям для подальшої обробки, є важливою задачею в обробці великих обсягів документів різного виду. У роботах [12,13] розглядаються методи машинного навчання та глибинного навчання, що використовуються для класифікації сканованих документів за різними характеристиками, такими як печатки [12], логотипи, друковані текстові блоки, підписи та таблиці [13]. Автори в роботах [14,15] обговорюють застосування класифікації документів у конкретних сферах, таких як акредитація та юридичні документи, де використання глибинних технологій допомагає підвищити точність і швидкість обробки. У роботах [16,17] представлено застосування методів штучного інтелекту для класифікації технічних та наукових документів, що демонструє їх ефективність у специфічних наукових та професійних сферах. Авторами [18,19] розглядаються методи класифікації

документів за допомогою новітніх технологій комп'ютерного зору, таких як трансформери та крос-модальні глибинні мережі. У роботах [20,21] обговорюється класифікація документів з мультимодальним підходом, який комбінує текстові та візуальні зображення інформації для покращення результатів класифікації документів зі складними графічними елементами або комбінованими даними.

Важливим аспектом для визначення класу документу, що буде перевірятися відіграє аналіз макету документу. Макет дозволяє зрозуміти структуру документа, визначити його компоненти та прискорити процес його класифікації. У роботах [22,24] розглядаються методи та підходи до аналізу макету документів. Автор [22] описує основні підходи до аналізу макету документів, які дозволяють ідентифікувати структуру тексту, графіки, таблиці та інші елементи та обговорює два нещодавно розроблених методи для вирішення зазначеної проблеми. У роботі [23] презентовано платформу LayoutParser, яка є єдиним інструментом для глибокого навчання, що дозволяє виконувати аналіз зображень документів з різними макетами та елементами, пропонуючи потужні можливості для розпізнавання та аналізу документів у різних форматах. У статті [24] розглянуто комплексну систему VINYAS для аналізу макетів документів, яка використовує передові методи для розпізнавання складних структур таких документів, як журнали, газети та інші публікації. Система сприяє підвищенню точності та ефективності аналізу документів з складними макетами.

Наступним етапом після визначення виду документу є обробка тексту документів. У роботах [25,26] розглядаються новітні підходи до вилучення тексту. Огляд технології OCR в роботі [26] підкреслює її значення як основи для сучасної інтелектуальної автоматизації, зокрема у процесах вилучення тексту з документів. Автори в статті [25] пропонують гібридну глибоку нейронну мережу з адаптивною оптимізацією галактичного рою для вилучення тексту з сценічних зображень, Дослідження у роботах [27,28] фокусується на досягненнях в OCR за допомогою глибокого навчання, пропонуючи алгоритми, які покращують точність розпізнавання тексту.

У контексті обробки рукописного тексту автори [29,30] звертаються до використання гібридних підходів на основі штучного інтелекту для поліпшення точності розпізнавання через покращення попередньої обробки зображень, що значно підвищує результативність при роботі з нечіткими чи складними зразками рукописного тексту.

Подальшим важливим етапом є класифікація та кластеризація тексту документу. У роботах [31, 32] представлено огляд та методи класифікації тексту через використання глибинного навчання та автоенкодерів. Робота [31] пропонує всебічний огляд підходів до класифікації текстів, що базуються на глибинному навчанні, а робота [32] фокусується на застосуванні попередньо навчених моделей для кластеризації текстів, що підвищує ефективність автоматизованого сортування та аналізу документів. Потужними інструментами для розуміння змісту тексту володіє така галузь як обробка природної мови NLP. У роботах [33, 34] розглядаються сучасні методи NLP у контексті глибинного навчання, автор [33] надає детальний огляд основних методів обробки природної мови, автор [34] використовує інтеграції глибинного навчання в NLP для створення потужних моделей, здатних до ефективної обробки текстової інформації в складних документних системах.

Верифікація документів є критичним етапом у процесі перевірки автентичності та цілісності документів. У роботах [35, 36] досліджуються підходи до верифікації підпису. В статті [35] автори пропонують методи офлайн перевірки рукописних підписів за допомогою синтетичної генерації підписів і використання сіамських нейронних мереж SNN, які, завдяки використанню ідентичних підмереж, дозволяють значно підвищити точність перевірки підписів у різних умовах. У роботі [36] розглядаються методи виявлення зміненого рукописного тексту в зображеннях документів за допомогою глибокого навчання, що сприяє вдосконаленню автоматичних систем перевірки документів на предмет підробок.

Верифікація особи (identity verification) є ще однією важливою частиною верифікації документів, де використовуються методи машинного навчання. Робота [37] надає приклад застосування машинного навчання для верифікації

ідентифікаційних документів в умовах, де доступ до контрольованих джерел інформації обмежений. Для виявлення змін у документах, що можуть бути ознакою підробки чи маніпуляції з інформацією, використовуються різні методи. У роботах [38, 39] представлені підходи до виявлення маніпуляцій із документами. Автори [38] використовують методи машинного навчання для виявлення маніпуляцій з документами та перевірки їхньої автентичності, а у роботі [39] автори пропонують огляд технік виявлення змін у зображеннях документів, які дозволяє забезпечити точність та надійність автоматизованих систем верифікації документів.

1.2 Аналіз існуючих методів, моделей та алгоритмів

Розглянемо методи, моделі та алгоритми, що використовуються на кожному етапі процесу автоматизованої перевірки документів.

Захоплення та оцифрування документів. Перший важливий етапом автоматизованої обробки документів. Він забезпечує перетворення документів з паперової (фізичної) форми або неструктурованої цифрової форми (зображення) у формат, який буде придатним для подальшого інтелектуального аналізу.

До методів захоплення документів відносять наступні: сканування, фотографування, отримання електронною поштою та імпорт (файлових систем, баз даних, тощо). Планшетні сканери підходять для невеликих об'ємів документів. Поточкові сканери дозволяють виконувати швидке сканування з автоматичною подачею, однак вони є значно дорожчими. Для певних задач та ситуацій можуть бути використані мобільні сканери – як пристрої, або як додатки для мобільних пристроїв. Вони дозволяють сканувати документи за допомогою камери мобільного пристрою. Таке сканування, як і метод фотографування документів за допомогою цифрових камер або камер смартфонів, є швидким та зручним для невеликих обсягів документів. Однак, якість зображень, що отримано шляхом фотографування, не є стабільною. Вона залежить від багатьох факторів, як освітлення, кут, стабільність, тощо. Зображення низької якості можуть потребувати значної попередньої обробки, щоб отримати читабельний текст та якісне

зображення графічних елементів. Отримання через електронну пошту та імпорт з зовнішніх сховищ є найбільш зручними, надійними та дешевими методами отримання зображень документів. Такі документи можуть бути представлені у різних форматах, тож отримані зображення можуть потребувати попередньої стандартизації та переведенню до одного формату. Основні алгоритми, які використовуються на цьому етапі стосуються лише обробки зображень та оптимізації процесу їх захоплення. Розглянемо популярні моделі для оцифрованих документів.

1. Растрові зображення – складаються з пікселів (прикладом можуть слугувати формати TIFF, JPEG, PNG). Вони мають великі розміри файлів, складні для редагування та масштабування без втрати якості. Такі зображення потребують додаткової обробки для розпізнавання тексту.

2. Векторні зображення описуються за допомогою геометричних фігур (прикладом може слугувати формат SVG). Такі зображення мають малий розмір файлів, легко редагуються та масштабуються без втрати якості. Однак, векторні зображення мають доволі обмежене застосування для текстових документів;

3. Формат PDF – гібридний, найбільш універсальний формат для зображень. Він підтримує текст, графічні елементи, векторну графіку, зберігає форматування та стиснення. Недоліком його є складності до редагування без спеціалізованого програмного забезпечення та автоматизованого аналізу без попередньої обробки.

Попередня обробка зображення (Image Preprocessing). Другий важливий етап у процесі автоматизованої перевірки документів. Попередню обробку використовують у випадках, коли вхідні документи уявляють собою скановані копії або фотографії похідних документів. Основна мета попередньої обробки – це покращення якості зображення. Висока якість зображення значно спрощує та підвищує точності подальшої обробки документа. Зазвичай, на наступних етапах використовуються оптичне розпізнавання символів OCR та аналіз макету документа.

До основних методів попередньої обробки зображень відносять наступні.

1. Видалення шумів, або Noise Reduction. В результаті видалення зайвих

шумів покращується якість зображення. Текст та інші елементи на зображенні стають «чіткішими» для OCR та подальшого аналізу. Однак невірно підібрані параметри шумоподавлення можуть навпаки, розмити дрібні деталі зображення (наприклад, тонкі лінії, дрібні символи, тощо). Можуть створити на зображенні спотворення, або, так звані, «артефакти». До того ж, деякі шумоподавляючі алгоритми є обчислювально складними та повільними.

2. Корекція освітлення та контрастності, або Illumination and Contrast Correction. Після застосування такого роду обробки на зображеннях покращується читабельність тексту. Однак, в результаті може бути спотворено оригінальні кольори та/або інтенсивності зображення. Це є критичним для зображень документів, які використовуються для перевірки їх автентичності;

3. Вирівнювання кута нахилу, або Skew Correction. В результаті вирівнювання точність оптичного розпізнавання символів підвищується та спрощується аналіз структури документу. Однак, невірне визначення кута нахилу документу на зображенні може призвести до певних спотворень. Такі спотворення можуть бути недопустимими для перевірки документу зі складним макетом.

4. Видалення спотворень, або Distortion Correction. Завдяки такій попередній обробці точність OCR підвищується. З іншого боку, такі алгоритми є складними та потребують точних налаштувань. В іншому випадку буде велика можливість втрати інформації або створення штучних спотворень.

5. Бінаризація зображення, або Binarization, або переведення зображення у біле-чорне. Таку обробку застосовується до зображень у градаціях сірого, або для кольорових зображень, де колір не є важливим елементом для перевірки. Бінаризація зменшує обсяг даних для подальшої обробки та покращує контраст між текстом та фоном, що, в свою чергу, є дуже важливим для OCR. Однак, тут теж важливим моментом є правильний вибір порогу бінаризації: точки якої яскравості відносити до білого, а якого до чорного. Помилковий поріг може призвести до втрати дрібних деталей тексту або навпаки, їх збільшення за рахунок об'єднання близьких.

6. Видалення плям та артефактів, або Spot and Artifact Removal. Після такої

попередньої обробки шум на зображенні зменшується. Це запобігає помилковому розпізнаванню як тексту або інших елементів усіляких плям та артефактів на зображенні. Однак, цей вид обробки також потребує ретельного налаштування параметрів. Помилкові параметри призведуть до втрати дрібних, але важливих, деталей документу (наприклад, крапки, дрібні символи, тощо).

7. Сегментація зображення, або Image Segmentation. Сегментація документа розділяє його на окремі області (наприклад, текст, таблиці, зображення) для їхньої подальшої незалежної обробки. Застосування сегментації покращує точність вилучення даних зі структурованих елементів. Однак, сегментація складних макетів може бути доволі складною за обчисленням. Алгоритми глибоко навчання, що використовуються для сегментації, вимагають значних обсягів розмічених даних для навчання. До того ж, помилково визначені сегменти можуть надалі призвести до некоректного вилучення інформації.

Методи та алгоритми попередньої обробки зображень зазвичай використовуються послідовно. Множина та послідовність їх виконання, їх налаштування підбираються відповідно до результату, що треба отримати. Такі набори методів мають ще одну назву - моделі попередньої обробки зображень. Налаштування таких моделей можуть виконуватися вручну після аналізу вхідних даних, або автоматично, за допомогою алгоритмів вибору оптимальних параметрів. Також можливо налаштувати такі моделі на основі підходів машинного навчання. У цьому випадку передбачається, що систему треба попередньо натренувати на певних типах зображень. Використання штучного інтелекту на етапі попередньої обробки сьогодні вважається одним із найперспективніших напрямів, оскільки дозволяє суттєво підвищити якість оброблених зображень і забезпечити їх готовність до подальших етапів аналізу.

Основними методами попередньої обробки зображень з використанням ШІ можна вказати наступні.

1. Навчання на основі даних. Таке навчання передбачає, що моделі навчаються на великих наборах даних зображень. Існує два підходи до такого навчання:

- навчання з вчителем, або Supervised Learning: кожне зображення окрім багатьох версій з недоліками має відповідний «ідеальний» зразок, що дозволяє їй вивчати складні закономірності та адаптуватися до різних типів та ступенів недоліків на зображенні;
- навчання без вчителя, або Unsupervised Learning: модель самостійно знаходить структури на зображеннях.

2. Навчання з підкріпленням, або Reinforcement Learning. Таке навчання також передбачає попереднє навчання моделі, але за допомогою підкріплення. У якості підкріплення виступає покращення якості зображення, виміряне певними метриками або оцінками результатів подальшої обробки зображення.

3. Згорткові нейронні мережі CNN. Наразі є найпоширенішою та найпопулярнішою архітектурою глибокого навчання для обробки зображень. CNN можуть бути навчені багатьом різним завданням попередньої обробки, як то шумаподавлення, супер роздільність, видалення розмиття, корекція освітлення, тощо.

4. Рекурентні нейронні мережі RNN. Менш поширені за CNN щодо безпосередньої обробки зображень. Однак, можуть бути доволі корисними для обробки послідовностей зображень. Вони здатні гарно зменшувати часовий шум та стабілізувати зображення.

5. Генеративно-змагальні мережі GAN. Такі мережі складаються з двох нейронних мереж: генератора та дискримінатора: Генератор створює оброблені зображення, а дискримінатор намагається відрізнити їх від реальних «чистих» зображень. GAN використовують для складних завдань, як то відновлення деталей, підвищення роздільної здатності, видалення небажаних об'єктів, заповнення відсутніх частин зображення, тощо.

6. Автоенкодері, або Autoencoders. Це нейромережі, що мають на меті стиснення вхідних даних у компактне представлення, а потім відновлення цих даних назад до їх оригінального вигляду, які використовуються для очищення зображень (видалення шуму, виявлення аномалій) за рахунок поганого відновлювання аномальних чи незнайомих даних.

У таблиці 1.1 наведено перелік найпоширеніших моделей які використовуються для обробки зображень за допомогою машинного навчання.

Таблиця 1.1

Моделі машинного навчання для обробки зображень

Назва моделі	Призначення	Базова модель	Метод навчання
DnCNN	Видалення шуму	CNN	Supervised Learning
Autoencoders (DAE)	Очищення зображень	AE	Unsupervised Learning
GANs (CycleGAN, Noise2Noise)	Фільтрація шуму	GAN	Unsupervised Learning
EnlightenGAN, Zero-DCE	Покращення освітлення	GAN	Unsupervised Learning
DocUNet, DewarpNet	Вирівнювання документів	CNN	Supervised Learning
U-Net, DeepLabV3+	Сегментація зображень	CNN	Supervised Learning
SRCNN, ESRGAN, SR3	Суперроздільність	GAN, CNN	Supervised Learning
DeepFill v2, CA-GAN	Відновлення зображень	GAN	Supervised Learning

Оптичне розпізнавання тексту – це процес переведення зображень тексту (рукописного або друкованого) у текст, який можна буде обробляти за допомогою комп'ютера. На сьогодні OCR уявляє собою вже цілу технологію, яка перетворює зображення тексту на машинний текст. Розпізнавання тексту є дуже важливим, бо саме його результати дозволяють подальшим алгоритмам аналізувати текстову інформацію в документах. Основними етапами розпізнавання тексту, окрім вказаної вже вище попередньої обробки, є наступні.

1. Вилучення ознак, або Feature Extraction – визначення характерних особливостей кожного символу, які відрізняють його від інших.

2. Класифікація символів, або Character Classification – ідентифікація кожного символу на основі вилучених ознак;

3. Постобробка, або Post-processing – виправлення помилок розпізнавання та покращення точності тексту.

Розглянемо основні моделі та алгоритми оптичного розпізнавання тексту. До появи нейромереж та мереж глибокого навчання розпізнавання тексту проводили за допомогою двох підходів. Перший – це зіставлення з шаблоном, або *Template Matching*. Підхід передбачає, що зображення кожного символу порівнюється з зображеннями у наборах попередньо визначених шаблонів. Він є дуже чутливим до типу шрифту, його розміру та спотворень. Другий – це вилучення ознак (*Feature Extraction*), коли допомогою структурний аналізу та статистичних методів вилучаються певні структурні ознаки символів (лінії, криві, петлі) та використовуються для класифікації. В сучасних підходах до оптичного розпізнавання символів вирізняють оптичне розпізнавання символів на основі машинного навчання *ML-based OCR* та оптичне розпізнавання символів на основі штучного інтелекту *AI-based OCR*. Машинне навчання є частиною штучного інтелекту, оскільки використовує алгоритми, які навчаються на даних і приймають рішення без жорстких правил. Штучний інтелект, в свою чергу, включає як традиційне машинне навчання, так і глибоке навчання *DL*, яке також базується на нейронних мережах, але для глибокого навчання використовуються багатопшарові нейронні мережі. Наразі для оптичного розпізнавання тексту з використанням штучного інтелекту використовуються як моделі на основі машинного навчання *ML-based OCR* так і моделі на основі глибокого навчання *DL-based OCR*.

Моделі на основі машинного навчання. Найчастіше такі моделі використовувалися в ранніх системах *OCR* та для певних специфічних завдань. Точність таких моделей може бути нижчою порівняно з моделями на основі глибокого навчання. Особливо ця тенденція проглядається при обробці складних зображень, рукописного тексту або текстів з різними шрифтами та стилями. Також ці моделі вимагають ретельного ручного розроблення ознак, що може бути складним та залежним від конкретного завдання. Тим не менш, моделі на основі машинного навчання успішно використовуються у системах з обмеженими обчислювальними ресурсами або невеликими наборами даних. Основними методами оптичного розпізнавання тексту на основі машинного навчання можна вказати наступні.

1. Метод опорних векторів SVM. Алгоритм навчання з учителем, який використовується для класифікації окремих символів на основі вилучених ознак. Алгоритм добре працює з великими наборами даних та є ефективним у високорозмірних просторах. Недоліками методу є необхідність ретельного вибору для нього ознак та велика обчислювальна складність для великих наборів даних.

2. Метод k-найближчих сусідів k-NN. Достатньо простий та легкий у реалізації алгоритмом, який класифікує символ на основі класу більшості його k-найближчих сусідів у просторі ознак. Алгоритм не потребує тривалого навчання, але є доволі чутливий до вибору k та метрики відстані, може бути повільним на етапі класифікації.

3. Наївний Баєсівський класифікатор, або Naive Bayes Classifier. Уявляє собою ймовірнісний класифікатор, заснований на застосуванні теореми Баєса з так званим «наївним» припущенням про незалежність між ознаками. Перевагами класифікатора є його простота, швидкість у навчанні та класифікації. До суттєвих недоліків класифікатора можна віднести його припущення про незалежність ознак, яке доволі часто на реальних даних не виконується.

4. Дерева рішень, або Decision Trees, та випадковий ліс, або Random Forest. Ідея підходу лежить у побудові деревоподібної структури для прийняття рішень на основі ознак. Випадковий ліс, в свою чергу, є множиною таких дерев рішень. Використання лісу підвищує точність та стійкість алгоритму. Алгоритм, що легко інтерпретується для окремих дерев, також може обробляти і нелінійні залежності. Окремі дерева можуть бути схильними до перенавчання, а випадковий ліс, навпаки, показує значно меншу схильність до перенавчання. Однак, випадковий ліс може виявитися обчислювально складним.

5. Приховані марковські моделі, або Hidden Markov Models. НММ моделі є статистичними моделями, які використовуються для моделювання послідовностей. Вони добре підходять для розпізнавання та моделювання послідовностей символів, з урахуванням контексту сусідніх символів. Однак, вони вимагають специфікованого визначення структури та її параметрів. До того ж такі моделі є доволі складними у навчанні.

Моделі основі глибинного навчання. У сучасних системах автоматизованої та інтелектуальної перевірки документів глибоке навчання є домінуючим підходом для розпізнавання тексту, завдяки його здатності автоматично вивчати ознаки та досягати у цьому високої точності.

1. Моделі на згорткових нейромережах CNN. Такі моделі найчастіше використовуються для оптичного розпізнавання символів. Вони здатні автоматично вивчати ієрархічні ознаки тексту з зображень та вилучати такі ознаки на рівні слів або навіть рядків. Прикладами архітектур для класифікації окремих символів можуть слугувати LeNet, AlexNet, VGG та ResNet.

2. Моделі на рекурентних нейромережах RNN. Ці моделі дозволяють враховувати залежності між сусідніми символами, що покращує точність розпізнавання. Такі рекурентні нейромережі як LSTM та GRU використовуються безпосередньо для моделювання послідовностей. Це є особливо важливим для коректного розпізнавання символів у контексті слова або рядка.

3. Комбіновані моделі нейромереж CNN + RNN. У багатьох сучасних системах оптичного розпізнавання тексту використовується комбінація згорткової нейромережі для вилучення ознак з зображення рядка та рекурентної для послідовного розпізнавання символів. Прикладом може слугувати архітектура згорткових рекурентних мереж CRNN, яка поєднує у собі обидва види нейромереж.

4. Моделі на основі уваги, або Attention Mechanisms. Механізми уваги дозволяють моделі фокусуватися на найбільш важливих частинах вхідного зображення під час розпізнавання. Для покращення розпізнавання довгих слів або текстів зі складним форматуванням механізм уваги застосовують в поєднанні з рекурентними нейромережами та/або трансформерами.

5. Трансформерні моделі, або Transformer Networks. Революційний надсучасний тип нейронних мереж у галузі обробки природньої мови, які також знайшли широке застосування в оптичному розпізнаванні безпосередньо тексту, текстів зі складним макетом або тексту кількома іноземними мовами. На відміну від рекурентних нейронних мереж, що обробляють послідовності даних послідовно, трансформери використовують механізм уваги та дозволяють моделі

одночасно враховувати всі елементи вхідної послідовності текстових даних. Це поєднання здатне усунути проблему «забування» інформації на довгих послідовностях, яка часто виникає в рекурентних нейромережах. А також дозволяє ефективніше моделювати залежності між віддаленими елементами. Ефективне моделювання контексту між символами та словами допомагає у ефективному розпізнаванні неоднозначних символів або слів. Також деякі моделі використовують трансформери для виконання всього процесу оптичного розпізнавання тексту наскрізна, від виявлення тексту до його розпізнавання, без явного розділення на окремі етапи. Завдяки здатності моделювати глобальні залежності, трансформери можуть краще обробляти документи зі складним макетом, де текст розташований у колонках, таблицях або перекриває зображення. Прикладами використання трансформерів може служити архітектура Vision Transformer (ViT), що була спеціально розроблена для комп'ютерного зору та розпізнавання тексту. Однак, є і недоліки. Трансформери можуть бути обчислювально складними, особливо для обробки зображень високої роздільної здатності. Також, навчання таких моделей вимагає великих обсягів розмічених даних.

Окремою задачею оптичного розпізнавання тексту є задача розпізнавання рукописного тексту. Для розпізнавання рукописних символів і текстів застосовуються різноманітні методи штучного інтелекту: моделі глибинного навчання на згорткових, рекурентних, згортковорекурентних нейронних мережах, графові нейронні мережі GNN та генеративні нейромережеві моделі GAN, алгоритм оптичного розпізнавання Tesseract, моделі, що використовують механізми уваги та моделі на основі трансформерів BERT, GPT. В задаче розпізнавання рукописного тексту широко використовуються наступні алгоритми:

- алгоритми розширення набору даних (Data augmentation), які дозволяють збільшити різноманітність зображень рукописного тексту і покращити здатність моделей до генералізації;
- алгоритми попередньої обробки зображень (Preprocessing);
- алгоритми постобробки результатів (Postprocessing), які використовують

мовні моделі або алгоритми корекції помилок та після первинного розпізнавання тексту можуть застосовуватись алгоритми для виправлення орфографічних помилок або коригування порядку слів.

У таблиці 1.2 наведено підходи та моделі, які використовуються для вирішення задач оптичного розпізнавання тексту за допомогою штучного інтелекту.

Таблиця 1.2

Підходи та моделі для вирішення задач оптичного розпізнавання тексту за допомогою штучного інтелекту

Задача	Машинне навчання (ML)	Глибинне навчання (DL)	Мовні моделі
Вилучення ознак (Feature Extraction)	SVM, k-NN, Decision Trees, Random Forest	CNN, Attention Mechanisms	-
Класифікація символів (Character Classification)	SVM, Naive Bayes, HMM, k-NN	LeNet, AlexNet, VGG, ResNet, CNN, Attention Mechanisms	-
Розпізнавання рукописного тексту (та постобробка)	Tesseract	CNN, RNN, CRNN, Transformer (ViT), GAN, Attention Mechanisms, Transformer	BERT, GPT
Розпізнавання тексту в зображеннях	Tesseract	LeNet, AlexNet, VGG, ResNet, CNN, Vision Transformer (ViT)	-
Моделювання послідовностей	HMM	LSTM, GRU, RNN, Transformer	BERT, GPT
Визначення тексту в складних зображеннях	Tesseract	CNN, Vision Transformer (ViT)	-
Обробка тексту з різними шрифтами та стилями	SVM, Naive Bayes	CNN, CRNN, Vision Transformer (ViT)	-

Аналіз макета документа є етапом автоматизованої перевірки документів, що дозволяє системам розуміти структуру та організацію інформації. Розглянемо ряд методів та алгоритмів штучного інтелекту, що дозволяють реалізувати аналіз макета документа.

1. Методи комп'ютерного зору, або Computer Vision, використовуються для вирішення наступних задач:

- виявлення об'єктів, або Object Detection, виконується за допомогою таких популярних алгоритмів як YOLO, Faster R-CNN та SSD, що навчаються на великих наборах даних розмічених документів та

дозволяють виконувати виявлення та локалізацію елементів макета (таблиці, поля, підписи, печатки, тощо);

- сегментація зображень, або Image Segmentation, виконується за допомогою методів U-Net та Mask R-CNN для розділення документа на окремі зони або об'єкти;
- аналіз геометричних властивостей реалізується згортковими та графовими неймережами, алгоритмами виявлення аномалій Isolation Forest, One-Class SVM та зіставлення шаблонів Template Matching, методами кластеризації та регресійного аналізу за допомогою машинного навчання та цілим рядом алгоритмів обробки зображень;

2. Методи обробки природної мови NLP використовуються для вирішення таких задач як аналіз текстових блоків та вилучення інформації. Використання BERT моделей дозволяє розуміти семантику та контекст тексту, а розпізнавання іменованих сутностей NER, дозволяє вилучати відносини та аналізувати залежності.

3. Методи глибокого навчання, або Deep Learning, дозволяють класифікувати документ за типом на основі його макету за допомогою згорткових неймереж, враховувати складні взаємозв'язки між елементами макета за допомогою графових неймереж та визначати логічну структуру документа за допомогою рекурентних неймереж та трансформерів.

4. Гібридні методи використовують комбінацію методів комп'ютерного зору CV та методів обробки природної мови NLP для виявлення візуальних елементів та аналізу текстових блоків з метою більш точного визначення структури документа. Також для обробки документів з варіативним макетом гібридні методи використовують правила та шаблони комбіновані з методами ШІ.

5. Алгоритми кластеризації дозволяють групувати схожі документи на основі їх макета для автоматичної класифікації документів. В таких алгоритмах підходи ШІ використовуються для автоматичного вилучення ознак, нелінійної та адаптивної кластеризації та кластеризації на основі семантики.

6. Алгоритми виявлення аномалій дозволяють виявляти відхилення від очікуваного макета. Відхилення від макету може свідчити про підробку документу, або помилки у ньому. Алгоритми виявлення аномалій можуть використовувати підходи ШІ для адаптивного виявлення аномалій, виявлення складних аномалій або аномалій на основі контексту. Вони можуть включати аналіз статистичних властивостей макета та використовувати моделі машинного навчання для виявлення незвичайних патернів.

У таблиці 1.3 наведено підходи та моделі, які використовуються для вирішення задач аналізу макету документу за допомогою штучного інтелекту.

Таблиця 1.3

Підходи та моделі для вирішення задач аналізу макету документу за допомогою штучного інтелекту

Задача	Машинне навчання (ML)	Глибинне навчання (DL)	Мовні моделі
Виявлення об'єктів (Object Detection)	-	Faster R-CNN, YOLO, SSD	-
Сегментація зображень (Image Segmentation)	-	Mask R-CNN,U-Net	-
Аналіз геометричних властивостей	Isolation Forest, One-Class SVM	GNN, RNN	-
Аналіз текстових блоків	-	-	BERT, RoBERTa
Вилучення іменованих сутностей (NER)	-	-	BERT, RoBERTa
Вилучення відносин (Relation Extraction)	-	-	BERT, RoBERTa
Аналіз семантики та контексту	-	-	BERT, GPT
Парсинг залежностей (Dependency Parsing)	-	-	BERT, GPT
Класифікація документа за типом	-	CNN,RNN, Transformer	-
Визначення логічної структури документа	-	RNN, GNN	-
Вилучення структурованої інформації (CRF, шаблони)	CRF, Template-based, Rule-based	-	-
Вилучення таблиць	-	Computer Vision + NLP	-

Вилучення структурованої та неструктурованої інформації з документів є етапом автоматизованої перевірки документів, що дозволяє системам отримувати необхідні дані для подальшого аналізу та валідації. Розглянемо ряд методів, моделей та алгоритмів штучного інтелекту, що дозволяють реалізувати вилучення структурованої та неструктурованої інформації з документів.

1. Методи обробки природної мови NLP для вилучення неструктурованої інформації.

- розпізнавання іменованих сутностей NER дозволяє виявляти та класифікувати іменовані сутності в тексті (імена, дати, організації, адреси тощо), використовує моделі BERT, RoBERTa та інші трансформерні моделі, що здатні розуміти контекст;
- вилучення відносин, або Relation Extraction, дозволяє виявляти зв'язки між іменованими сутностями (наприклад особа та її адреса, або організація та її місцезнаходження);
- аналіз семантики та контексту дозволяє вилучати інформацію, яка неявно міститься в тексті (контексту тексту) за допомогою аналізу тональності, тематичного моделювання, семантичний аналіз та інші;
- парсинг залежностей, або Dependency Parsing аналізує граматичну структуру речень, що допомагає виявити зв'язки між словами та фразами.

2. Методи машинного навчання для вилучення структурованої інформації.

- умовні випадкові поля CRF використовуються для вилучення структурованих даних з документів (таблиці, форми) та дозволяють враховувати контекст та залежності між полями даних;
- методи, засновані на шаблонах, або Template-based methods, використовують шаблони для вилучення даних з документів з відомою структурою (технічні паспорти, сертифікати, транспортні накладні, медичні карти, тощо);
- методи, засновані на правилах, або Rule-based methods використовують правила, визначені експертами, для вилучення

даних з документів та можуть бути корисними для обробки документів з чітко визначеними правилами.

3. Методи глибокого навчання для вилучення інформації.

- моделі для вилучення інформації, або Information Extraction models, використовують рекурентні RNN нейромережі та трансформерні моделі, які можуть бути навчені виявляти складні залежності між даними та обробляти документи з варіативною структурою;
- моделі для вилучення табличних даних, або Table Extraction models, є спеціалізовані моделями, навченими для вилучення структури таблиць та дані з різних комірок, використовують рекурентні RNN, згорткові CNN мережі, трансформерні моделі BERT, T5 та Template-based methods машинного навчання;
- візуальне вилучення інформації, або Visual Information Extraction, поєднує методи комп'ютерного зору та NLP, які здатні обробляти зображення документів та скановані PDF-файли для вилучення інформації.

4. Гібридні методи можуть бути представлені наступними комбінаціями:

- методи NLP, що аналізують текстові блоки, та методи машинного навчання, що вилучають структуровані дані, для більш повного розуміння інформації в документах;
- правила та шаблони, що аналізують документів з відомою структурою, та методи ШІ, що аналізують документи з невідомою структурою, для підвищити точність та ефективність вилучення інформації.

Виявлення аномалій та потенційних ознак шахрайства є етапом автоматизованої перевірки документів, який дозволяє підвищити рівень безпеки, знизити ризики та адаптуватися до нових видів шахрайства. Такі методи штучного інтелекту, як машинне навчання та глибинне навчання, що використовуються для виявлення аномалій, було оглянуто вище при огляді методів розпізнавання тексту, аналізу макету та вилучення структурованої та неструктурованої інформації.

Оглянуті методи легко масштабуються до виявлення аномалій на рівні документу. Розглянемо інші підходи до виявлення ознак шахрайства при автоматизованій перевірці документів:

– аналіз візуальних ознак: для виявлення ознак підробки, таких як невідповідності шрифтів, незвичайні водяні знаки та зміни зображень можуть використовуватися методи комп'ютерного зору, а моделі глибинного навчання можуть бути навчені виявляти складні візуальні патерни, які можуть свідчити про можливе шахрайство;

– аналіз текстових ознак як то незвичайні фрази, граматичні помилки та невідповідності змісту, аномалії в стилі написання та семантиці тексту використовують методи обробки природної мови NLP;

– аналіз метаданих документів, таких як дата створення, автор та історія змін, може бути виявлено за допомогою простих моделей машинного навчання, які навчені виявляти незвичайні патерни в метаданих.

У таблиці 1.4 наведено підходи та моделі, які використовуються для вирішення задач вилучення структурованої та неструктурованої інформації з тексту за допомогою штучного інтелекту.

Таблиця 1.4

Підходи та моделі для вирішення задач вилучення структурованої та неструктурованої інформації документу за допомогою штучного інтелекту

Задача	Машинне навчання (ML)	Глибинне навчання (DL)	Мовні моделі
Вилучення структурованої інформації	CRF, Template-based, Rule-based	-	-
Вилучення неструктурованої інформації	-	-	BERT, RoBERTa
Вилучення іменованих сутностей (NER)	-	-	BERT, RoBERTa
Вилучення відносин (Relation Extraction)	-	-	BERT, RoBERTa
Аналіз семантики та контексту	-	-	BERT, GPT
Парсинг залежностей (Dependency Parsing)	-	-	BERT, GPT
Моделі для вилучення інформації (Information Extraction)	ML+NLP, Rule-based, Template-based	RNN, Transformer	-

Продовження таблиці 1.4

Підходи та моделі для вирішення задач вилучення структурованої та неструктурованої інформації документу за допомогою штучного інтелекту

Задача	Машинне навчання (ML)	Глибинне навчання (DL)	Мовні моделі
Моделі для вилучення табличних даних	Template based	RNN, GNN	BERT, T5
Візуальне вилучення інформації	-	Computer Vision + NLP	-
Виявлення аномалій в метаданих	SVM, Isolation Forest	-	-
Виявлення аномалій в тексті	-	-	GPT, BERT
Виявлення візуальних аномалій	-	CNN, GAN, Autoencoders, RNN	-
Аналіз текстових ознак	-	-	NLP (на основі трансформерів)
Аналіз візуальних ознак (аналіз шрифтів, водяних знаків)	-	CNN, GAN	

Автоматизована перевірка автентичності візуальних елементів документів є етапом автоматизованої перевірки, необхідним для перевірки документів, які містять підписи, печатки, водяні знаки та інші візуальні маркери. Розглянемо основні підходи для перевірки за допомогою штучного інтелекту:

1. Перевірка таких захисних елементів як: водяні знаки, мікрошрифти, гільйоши (хвилясті візерунки) виконується наступними методами:

- методи обробки зображень Fourier Transform, Wavelet Transform для виявлення водяних знаків, виявлення їх відсутності або підробки;
- генеративно-змагальні мережі GAN для генерації еталонних зразків і порівняння зразка, що обробляється з еталоном;
- нейромережі глибинного навчання ResNet, EfficientNet, Vision Transformers для автоматичної перевірки складних графічних елементів;
- алгоритм SIFT, особливістю якого інваріантність до масштабних змін, поворотів, освітлення та часткових спотворень, для знаходження характерних точок та візерунків на документах. для

виявлення та опису особливих точок (ключових ознак) та візерунків на зображенні документу

2. Перевірка голограм на та оптичних захисних елементів виконується для визначення справжності голографічних захисних елементів та виявлення відсутності або неправильної відбивної здатності голограм виконується наступними методами:

- тривимірні згорткові мережі 3D CNN та трансформери для аналізу зображень ViT використовуються для перевірки текстури, складних візуальних структур та світло відбивних властивостей голограм;
- аналіз спектра зображення, або Spectral Analysis, з згортковими неймережами CNN використовуються для аналізу світлового відбиття.
- поляризаційний аналіз, або Polarization Analysis, з моделями глибинного навчання ResNet, EfficientNet, ViT використовується для перевірки специфічних оптичних властивостей елементів.
- Виявлення слідів підробки, редагування, модифікації документа виконується наступними підходами:
- згорткові неймережі CNN та моделі ResNet, MobileNet використовуються для класифікації підроблених та справжніх документів, створених шляхом копіювання й вставлення частин документу;
- автоенкодери, а саме варіаційні автоенкодери Variational Autoencoders (VAE) та автоенкодери для видалення шуму Denoising Autoencoders (DAE), використовуються для виявлення аномалій у структурі документа, цифрового редагування фотографій та текстових областей;
- метод Error Level Analysis (ELA) з інтегрованими згортковими неймережами CNN використовується для виявлення змін у JPEG-зображеннях, які можуть вказувати на підробку;
- традиційні методи обробки зображень (для виявлення контурів і градієнтів на зображеннях) фільтри Sobel, Laplacian, Canny Edge

Detection з інтегрованими згортковими неймережами CNN використовуються для виявлення слідів редагування, ретушування або вирізання/замальовування (наприклад, Photoshop) у документі.

3. Розпізнавання та перевірка підписів, штампів та печаток спрямовано на виявлення фальшивих печаток, які не відповідають зразкам та визначення цифрових підробок або накладання штампів:

- мережі для пошуку схожестей SNN використовуються для порівняння підписів або печаток з еталонними, що дозволяє порівнювати підписи та печатки з базою зразків;
- графові неймережі GNN використовуються для аналізу структури підписів та рукописного тексту;
- виявлення характерних рис зображень (Histogram of Oriented Gradients, HOG) в комбінації з методом класифікації SVM використовуються для перевірки підроблених підписів або печаток.

4. Перевірка фотографій та біометричних даних спрямована на перевірку відповідності фото особи в документі реальній особі, виявлення змінених або штучно згенерованих фотографій та перевірку фотографій на відповідність правилам (фон, розмір, тощо):

- технологія Face Recognition, моделі Facenet, ArcFace та бібліотеки Dlib, OpenCV з інтегрованими неймережами глибинного навчання використовуються для порівняння фото в документі з фотографією особи;
- глибинне навчання використовується у технологіях виявлення живості Liveness Detection, що перевіряє, чи є особа, яка подає своє обличчя для розпізнавання, реальною живою людиною, а не статичним зображенням або відео, 3D-реконструкції обличчя (3D Face Reconstruction), що створює тривимірні моделі обличчя на основі двовимірних зображень або відео, та методі виявлення підроблених зображень Spoofing Detection CNNs, які, в свою чергу, використовуються для виявлення підроблених фотографій або сканів;

- технологія GANs Detector на основі генеративно-змагальні мереж використовується для виявлення фото, створених нейромережами.

У таблиці 1.5 наведено підходи та моделі, які використовуються для вирішення задач вилучення структурованої та неструктурованої інформації з тексту за допомогою штучного інтелекту.

Таблиця 1.5

Підходи та моделі для вирішення задач автоматизованої перевірки автентичності візуальних елементів документів за допомогою штучного інтелекту

Задача	Машинне навчання (ML)	Глибинне навчання (DL)
Перевірка водяних знаків, мікрошрифтів, гільйонів	-	GAN, ResNet, EfficientNet, Vision Transformers
Перевірка голограм та оптичних елементів	-	3D CNN, ViT, ResNet, EfficientNet
Виявлення слідів підробки або редагування документа	-	CNN, VAE, DAE, Error Level Analysis
Перевірка підписів, печаток та штампів	Siamese Networks, SVM	GNN, CNN
Перевірка фотографій та біометричних даних	-	FaceNet, ArcFace, Dlib, Liveness Detection, Spoofing Detection, GANs Detector

1.3 Аналіз сучасних засобів програмної інженерії для автоматизованої перевірки документів

Розглянемо популярні інструменти для обробки документів на базі ШІ та оптичного розпізнавання символів OCR.

Інструменти Google Cloud Document AI, Microsoft Azure AI Document Intelligence та Amazon Textract уявляють собою частини хмарних платформ, які надають різноманітні сервіси для обробки документів.

Google Cloud Document AI підтримує шаблони документів (custom processors), аналіз структури (layout analysis), витягування полів за шаблонами. Підходить для перевірки форм звітності, договорів, накладних тощо. Характеризується високою точністю для друкованих документів та частково

підтримує рукописного тексту, перевірку захисних елементів документів підтримує на базовому рівні та обмежено підтримує українську мову [40].

Microsoft Azure AI Document Intelligence (Form Recognizer) підтримує користувацькі шаблони документів, витяг даних, верифікацію, структурування документу, рукописний текст та формат PDF. Підходить для корпоративної документації з фіксованим форматом, як то акти, звіти, технічні форми, тощо. Надає високу точність обробки шаблонних документів, часткова підтримує рукописний текст, підтримка захисних елементів у продукті на базовому рівні. Достатньо добре підтримує українську мову [41].

Amazon Textract разом з Amazon Comprehend підтримує витяг тексту, таблиць і структурованих даних, а також класифікацію документів. Дозволяє налаштування шаблонів та правил перевірки. Підходить для бухгалтерських та фінансових документів з чітким форматом. Характеризується високою точністю для витягування тексту і структурованих даних, обмеженою підтримкою української мови, обмеженою підтримкою рукописного тексту, базовою підтримкою захисних елементів [42].

Наступні продукти поставляються як самостійні програмні рішення або системи, що можуть бути інтегровані в локальні або приватні хмарні середовища.

Нуратос підтримує високоточну обробку документів на основі шаблонів та налаштування під внутрішні документи компанії. Підходить для перевірки реквізитів і структурних полів у внутрішніх шаблонах. Має високу точність для шаблонних документів, обмежену підтримку української мови, часткову підтримку рукописного тексту, базову підтримку захисних елементів [43].

Abbyy Vantage та FlexiCapture підтримує шаблонне розпізнавання, структурний аналіз, логічну перевірку полів та розширене налаштування шаблонів під документи компанії. Підходить для перевірки внутрішніх документів з чіткою структурою, таких як технічні акти, заявки, таблиці та довідки. Висока точність для шаблонних документів, підтримка української мови, часткова підтримка рукописного тексту, розширена підтримка захисних елементів [44,45].

Rossum AI підтримує "Cognitive data capture" – витяг без необхідності жорсткого шаблону. Після навчання адаптується під користувацькі документи та дозволяє створювати правила перевірки даних. Гарно підходить для напівструктурованих документів, таких як службові записки та складські форми. Має високу точність для напівструктурованих документів, обмежену підтримку української мови, обмежену підтримку рукописного тексту, базову підтримку захисних елементів [46].

Klipra DocHorizon підтримує OCR, шаблонне розпізнавання та класифікацію документів. Це API-first платформа для інтеграції в внутрішні системи з можливістю валідації даних. Підходить для HR-документації, звітів та комерційних пропозицій. Має високу точність для OCR і шаблонних документів, обмежену підтримку української мови, часткову підтримку рукописного тексту, базову підтримку захисних елементів [47].

Nanonets підтримує тренування на власних шаблонах документів та легку інтеграцію через API. Виявляє аномалії у заповненні шаблонів. Підходить для шаблонних звітів, анкет, заяв та договорів. Має високу точність для шаблонних документів, обмежену підтримку української мови, часткову підтримку рукописного тексту, базову підтримку захисних елементів [48].

Розглянемо інструменти, що надають можливості налаштування механізмів перевірки документів за допомогою штучного інтелекту та особливості їх використання на прикладі Google Cloud Vision API.

Google Cloud Vision API [49] – інструмент, частина хмарних сервісів Google, який надає розробникам можливість використовувати попередньо навчені моделі машинного навчання для аналізу зображень. Для автоматизованої перевірки документів та документації інструмент може бути використано у наступних цілях.

- виявлення та класифікація об'єктів, що присутні на зображенні.
- вилучення тексту з зображень;
- розпізнавання рукописного тексту;
- виявлення обличчя на зображеннях, а також визначення емоцій та інших характеристик;

- виявлення логотипів компаній на зображеннях;
- визначення загальних властивостей зображення, таких як кольори та освітлення.

Найпоширенішим цифровим форматом для документів є формат pdf. Cloud Vision API має обмежений функціонал у роботі з цим форматом, однак може виявляти та транскрибувати текст із PDF-файлів, що зберігаються в Google Cloud Storage. Така обробка PDF-файлів відбувається асинхронно: необхідно завантаження PDF-файл до Google Cloud Storage та запит до Vision API зі шляхом до файлу. Після обробки результати зберігаються у вказаному місці в Cloud Storage.

Google Cloud Vision API вже є попередньо навчений для розпізнавання загальних об'єктів і тексту, однак існує можливість розширити його функціональність, використовуючи інші сервіси Google Cloud Platform та методи машинного навчання для класифікації власних типів документів. Наприклад, OCR та Document AI перетворюють неструктуровані дані в структурований формат (текст, пари ключ-значення, таблиці), який можна використовувати для машинного навчання. Вилучений текст та дані можна використати як «ознаки» для моделей машинного навчання – ключові слова, фрази, значення з певних полів можуть використовуватися для визначення типу документа. Для створення та навчання моделей, які класифікують документи на основі вилучених даних, використовують Vertex AI — платформу машинного навчання (ML) від Google Cloud, яка надає інструменти та послуги для створення, навчання та розгортання моделей ML у хмарі. Для швидкого створення прототипів, налаштування та розгортання з генеративних моделей штучного інтелекту у складі платформи Google Cloud Vertex AI використовують інструмент Vertex AI Studio.

Подібний функціонал пропонує Microsoft Azure AI Vision [50] – частина хмарних сервісів Microsoft Azure, яка надає інструменти комп'ютерного зору (Computer Vision) для аналізу зображень і відео за допомогою штучного інтелекту. Пропонує функції OCR та аналізу зображень для отримання інформації з візуального контенту, що може бути використано для перевірки документів.

Функціонал щодо аналізу макету документу надають наступні інструменти:

- Python бібліотека LayoutParser використовує моделі машинного навчання для аналізу макету документів, дозволяє класифікувати такі елементи документа, як текст, таблиці, зображення;

- DocTR – інструмент для аналізу макетів документів, який використовує нейронні мережі для розпізнавання розміщення тексту та інших елементів у документах.

Перевірити та верифікувати ідентифікаційні документи можливо з використанням таких інструментів, як:

- Jumio AI Document Verification використовує штучний інтелект для автентифікації документів, забезпечуючи швидке та точне підтвердження особи;

- Veridas Document Verification застосовує AI для виявлення підроблених або змінених документів, підвищуючи рівень безпеки та захищаючи репутацію компаній;

- Klippa Document Verification комбінує OCR, машинне навчання та комп'ютерний зір для перевірки автентичності та дійсності документів.

Open Source бібліотеки:

- Tesseract OCR – бібліотека для оптичного розпізнавання символів.

- бібліотеки Python (TensorFlow, PyTorch, Scikit-learn) дозволяють створення власних моделей машинного навчання для класифікації та валідації документів.

- OpenCV – бібліотека для комп'ютерного зору, що надає широкий спектр алгоритмів для обробки зображень, яка може використовуватися як для аналізу макету документів та вилучення інформації, так і для виявлення та аналізу водяних знаків, голограм, мікро тексту та інших захисних елементів, надає інтерфейси для використання попередньо навчених моделей глибокого навчання, а також інструменти для навчання власних моделей.

2 РОЗРОБКА МЕТОДУ АВТОМАТИЗОВАНОЇ ПЕРЕВІРКИ СЕРТИФІКАЦІЙНИХ АВІАЦІЙНИХ ДОКУМЕНТІВ

2.1 Аналіз сертифікаційної авіаційної документації України

Система сертифікації авіаційної документації в Україні не розробляє унікальні національні форми документів, натомість в Україні йде процес гармонізації авіаційного законодавства з нормами Європейського Союзу, що дозволяє безпосередньо приймати міжнародні стандартизовані форми, що розроблено Агентством з безпеки польотів Європейського Союзу EASA. Це підтверджується наступними фактами та документами.

1. Угода про Спільний авіаційний простір (САП) була підписана 12.10.2021р та ратифікована 17.02.2022р. Угода зобов'язує Україну впроваджувати у своє законодавство відповідні норми та стандарти ЄС у сфері цивільної авіації, включаючи ті, що розроблені EASA і є наразі головним юридичним механізмом.[51].

2. Пряме використання форм EASA: при процедурі сертифікації персоналу з технічного обслуговування (Part-147) необхідно заповнення «Форми 4 EASA» (EASA Form 4) для керівного персоналу [52].

3. Визнання сертифікатів EASA. Авіаційні правила, що регулюють видачу свідоцтв льотним екіпажам демонструють, що сертифікати, видані за стандартами EASA, визнаються в Україні, усуваючи потребу в окремій національній сертифікації та інші [53].

На національному рівні механізмом імплементації європейських норм є Авіаційні правила України (АПУ), що видаються Державіаслужбою України на основі Повітряного кодексу України. Такі правила як Part-145 (Технічне обслуговування) або Part-21 (Сертифікація) є офіційним перекладом та імплементацією відповідних регламентів EASA в українське правове поле. Тобто,

документи, що видаються в Україні, є формою EASA, з мінімальною національною адаптацією [54].

2.1.1 Класифікація сертифікаційної авіаційної документації

Таким чином, класифікацію сертифікаційних авіаційних документів України будемо розглядати на номенклатурі форм EASA.

1. Сертифікація організацій:

- схвалення організації з конструювання Design Organisation Approval – це дозвіл компанії (наприклад, Airbus, Boeing) розробляти та модифікувати авіаційну техніку;
- схвалення організації з виробництва Production Organisation Approval – дозвіл компанії серійно виробляти авіаційну техніку відповідно до схваленого дизайну;
- сертифікат експлуатанта Air Operator Certificate – це дозвіл для авіакомпанії (наприклад, Turkish Airlines, Lufthansa) виконувати комерційні повітряні перевезення;
- схвалення організації з технічного обслуговування Maintenance Organisation Approval – MOA / Part-145, EASA Form 3 – це дозвіл для MRO-центрів (Maintenance, Repair, Overhaul) обслуговувати та ремонтувати повітряні судна.

2. Сертифікація типу, виробництва та початкової льотної придатності (АПУ-21/EASA Part-21) регулює сертифікацію конструкції повітряних суден, їх компонентів, схвалення організацій розробника DOA та виробника POA. Включає у себе наступні види сертифікатів:

- сертифікат типу Type Certificate (TC) видається на дизайн-проект повітряного судна або двигуна або його компонента (двигуна, гвинта, тощо) та підтверджує, що конструкція відповідає всім нормам льотної придатності (наприклад, CS-25 для великих пасажирських літаків в EASA), містить у якості додатку карту даних сертифіката типу Type Certificate Data Sheet (TCDS), яка по суті є

«свідоцтвом про народження» та «паспортом» для цілої модельної лінійки повітряних суден;

- сертифікат льотної придатності Certificate of Airworthiness (CofA, EASA Form 25) – це основний документ, що видається Державіаслужбою і засвідчує, що конкретне повітряне судно відповідає схваленій конструкції (сертифікату типу) та придатне до безпечної експлуатації;
- обмежений сертифікат льотної придатності Restricted Certificate of Airworthiness (RCofA, EASA Form 24) видається для повітряних суден, які не відповідають усім вимогам CofA, але були визнані безпечними для експлуатації за певних обмежень;
- заява про відповідність Aircraft Statement of Conformity (EASA Form 52) – це декларація схваленої організації з виробництва, яка засвідчує, що вироблене повітряне судно з конкретним серійним номером повністю відповідає схваленій конструкції та є передумовою для видачі первинного CofA;
- сертифікат передачі до експлуатації після виробництва Certificate of Release to Service видається організацією-розробником при передачі нового компонента (EASA Form 1) або повітряного судна (EASA Form 53) замовнику;
- сертифікат схвалення організації-виробника Production Organisation Approval Certificate (EASA Form 55).

3. Сертифікація підтримання льотної придатності (АПУ-М/EASA Part-M) регулює процеси управління постійною льотною придатністю повітряного судна:

- сертифікат перегляду льотної придатності Airworthiness Review Certificate (ARC, EASA Form 15b) – це ключовий документ, який підтверджує, що повітряне судно пройшло належний перегляд льотної придатності та є придатним до польотів на визначену дату, видається Державіаслужбою або, у певних випадках, схваленою

організацією з підтримання льотної придатності або організацією з технічного обслуговування;

- рекомендація щодо видачі ARC, Recommendation for ARC (EASA Form 15a) видається схваленою організацією з підтримання льотної придатності і подається до Державіаслужби як підстава для видачі ARC.

4. Сертифікація організацій з технічного обслуговування повітряних суден та компонентів (АПУ-145/EASA Part-145) регулює схвалення організацій, що виконують технічне обслуговування повітряних суден та їх компонентів:

- сертифікат дозволеної передачі Authorised Release Certificate (EASA Form 1) – це головний документ для компонентів повітряного судна, який засвідчує, що компонент (новий або після технічного обслуговування або ремонту) є придатним до встановлення на повітряне судно;
- сертифікат передачі до експлуатації Certificate of Release to Service (CRS) – це юридична заява, що вноситься у бортові журнали повітряного судна уповноваженим персоналом після завершення технічного обслуговування, що підтверджує, що все необхідне технічне обслуговування було виконано відповідно до АПУ-145 і повітряне судно в цілому є придатним до експлуатації;
- схвалення організації (Approval Certificate / Scope of Approval, EASA Form 2) – це частина загального сертифіката схвалення організації Part-145, де детально перераховано, які саме типи повітряних суден, двигунів або компонентів та які види робіт з технічного обслуговування організація має право виконувати.

5. Сертифікація персоналу:

- дані про керівний склад Management Personnel (EASA Form 4) – це форма, що використовується для подання до Державіаслужби інформації про ключовий керівний персонал (Post Holders) для їх

прийняття регулятором, є обов'язковою частиною процесу схвалення або підтримки схвалення будь-якої організації;

- ліцензії Licences – сертифікати для людей, ліцензії пілотів (PPL, CPL, ATPL), авіадиспетчерів та авіаційних техніків (Part-66).

Сертифікаційні авіаційні документи утворюють складну, але логічну структуру взаємопов'язаних правил. TCDS (дизайн) визначає, що було збудовано. POA контролює, хто це збудував. CofA підтверджує, що конкретний літак відповідає дизайну. AOC та Part-145 контролюють, хто і як його експлуатує та обслуговує.

Приклади сертифікаційних авіаційних документів наведено на рисунку 2.1.

2.1.2 Основні структурні елементи сертифікаційної авіаційної документації

За результатами аналізу основних сертифікаційних документів було визначено наступні основні структурні елементи сертифікаційних документів.

1. Ідентифікатор документу, що міститься у заголовку документу:

- назва органу, що видав документ – «European Union Aviation Safety Agency» або «Державна авіаційна служба України»;
- назва документа – «СЕРТИФІКАТ ТИПУ», «СЕРТИФІКАТ ЕКСПЛУАТАНТА», тощо;
- унікальний номер документа – «EASA.A.064» для Airbus A350 або
- «AOC UA.001».

2. Власник документу:

- назва організації або особи кому було видано цей документ – «Airbus Operations S.A.S.» або «ПрАТ Авіакомпанія»
- адреса власника – юридична та/або фактична адреса власника документу.



AOC



CoFA



Part-145



Part-21G

Рис. 2.1 Приклади сертифікаційних авіаційних документів

3. Об'єкт та сфера дії документу, відображає, що саме сертифікує документ:
- опис продукту або послуги – «Тип літака: A350-900» або «Виконання комерційних повітряних перевезень»;
 - перелік схвалених типів повітряного судна (для АОС/Part-145) показує перелік повітряних суден, які компанія може експлуатувати або обслуговувати – «Boeing 737-800», «Airbus A320neo», тощо;
 - географічні обмеження (для авіакомпаній) де компанія може використовувати повітряні судна – «Внутрішні та міжнародні польоти».
4. База сертифікації (підстава для видачі):
- перелік норм і правил – список конкретних стандартів, на відповідність яким проводилася перевірка, наприклад «EASA CS-25 Amendment 20» або «EU Regulation 965/2012 (Air OPS)»;
 - посилання на висновки та звіти – може містити посилання на документи, що підтверджують відповідність.
5. Умови та обмеження (The Fine Print):
- експлуатаційні обмеження – містить обмеження на експлуатацію, наприклад, «Тільки денні польоти», «Максимальна злітна вага», «Обмеження по рівню шуму», тощо;
 - вимоги до персоналу або обладнання, наприклад, «Наявність мінімум 2-х сертифікованих інженерів типу...»;
 - спеціальні умови (Special Conditions) – якщо технологія нова і для неї ще немає стандартних правил, регулятор прописує спеціальні умови безпеки.
6. Дати та підписи:
- дата видачі – коли документ набув чинності;
 - термін дії – може бути: «безстроковий» (Unlimited), «за умови дотримання умов» , або мати конкретну дату закінчення, після якої потрібна повторна сертифікація;

– підпис та печатка – підтвердження повноважною особою регуляторного органу.

2.1.3 Ключові ознаки сертифікаційних авіаційних документів

Зміст сертифікаційного документу регулюється міжнародними нормами, однак зовнішній вигляд документів залежить від органу, що їх видав (наприклад, EASA в Європі, FAA в США або Державіаслужба в Україні).

Проаналізуємо основні сертифікаційні документи з точки зору їх зовнішнього вигляду та елементів, як ключових ознак, що можуть бути використані для автоматизованої класифікації.

1. Логотип та назва регулятора. На рисунку 2.2 наведено логотипи наступних авіаційних агенцій та служб: EASA, FAA та Державіаслужба України.



Рис. 2.2. Логотипи авіаційних агенцій та служб

2. Заголовок документа. Текстовий елемент, що є ознакою виду сертифікату:

- «Air Operator Certificate» – «Сертифікат експлуатанта»;
- «Certificate of Airworthiness» – «Сертифікат льотної придатності»;
- «Maintenance Organisation Approval» – «Схвалення організації з ТО»;
- «Part-145», «Part-M», «Part-21» – можуть вживатися як частина назви.

3. Унікальний специфічний формат номера для сертифікатів:

- EASA Part-145: EASA.145.XXXX;

- EASA Part-21 (DOA/POA): EASA.21J.XXX (DOA) або EASA.21G.XXXX (POA);
- EASA AOC: [Код країни].AOC.XXX (напр., DE.AOC.123);
- Українські: UA.AOC.XXX, UA.145.XXXX, тощо.

4. Унікальний специфічні поля:

- поля серійного номеру «Manufacturer's Serial No.» та реєстраційного знаку «Registration Marks» присутні лише у сертифікаті льотної придатності (CofA);
- поле переліку типів повітряних суден «Aircraft Type(s)» та поле регіонів польоту «Area(s) of Operation» присутні лише у сертифікаті авіакомпанії-експлуатанта (AOC);
- поле категорії схвалення «Scope of Work» або «Rating» (які саме літаки/компоненти організація може обслуговувати) присутні лише у сертифікаті Part-145.

У таблиці 2.1 наведено ключові ознаки для автоматизованої класифікації документів.

Ключові ознаки верифікації сертифікаційних документів можна розділити на наступні 3 групи відповідно до складності їх ідентифікації простими засобами.

1. До групи ознак низької складності віднесемо ознаки, які можна побачити з неозброєним оком, які добре фіксуються сканерами чи камерами і добре піддаються автоматизації. Такими ознаками є: логотипи (EASA, FAA, Державіаслужба), чорнильні («мокрі») печатки, підписи та шаблони або фіксовані форми документів.

2. Ознаки середньої складності вимагають спеціального огляду або специфічних умов, які погано піддаються автоматизації. Такі ознаки можуть «загубитися» при простому скануванні чи захваті або фотографуванні камерою. До таких ознак віднесемо голограми, гільйоши (складні візерунки на папері документу) та мікродрук.

Таблиця 2.1

Таблиця ключових ознак для автоматизованої класифікації документів

Тип	Ключові слова у заголовку	Формат номеру	Унікальні поля	Графічні елементи
Сертифікат експлуатанта (АОС)	Air Operator Certificate Сертифікат експлуатанта	[Код країни].АОС.XXX UA.АОС.001	Area(s) of Operation (Регіони польотів) Aircraft Type(s) (Список типів ПС) Name of Operator	Логотип національного органу, може мати QR-код.
Схвалення організації з ТО (Part-145)	Maintenance Organisation Approval Схвалення організації з ТО Part-145	[Орган].145.XXXX EASA.145.1234	Таблиця «Scope of Work»/«Rating» (Категорії А, В, С, D)(частіше розміщено на 2-й сторінці)	Логотип EASA (якщо EASA Form 3), національного органу, може мати QR-код
Сертифікат льотної придатності (CofA)	Certificate of Airworthiness Сертифікат льотної придатності	Формат нестандартизований, може бути номер бланку	Registration Marks (Рєстраційний знак (UR-XYZ) Manufacturer's Serial No. (Серійний номер)	Захищений бланк (гільйош, водяні знаки), може мати тиснену печатку або голограму.
Схвалення організації з виробництва (Part-21G)	Production Organisation Approval Схвалення організації з виробництва Part-21G	[Орган].21G.XXXX EASA.21G.1234	«Scope of Production»/«Terms of Approval" (Список схвалених компонентів/ПС)	Логотип EASA (якщо EASA Form 55) або національного органу.
Сертифікат типу (ТС)	Type Certificate Сертифікат типу	[Орган].[Категорія].XXX EASA.A.064	Type Certificate Data Sheet (TCDS) (або посилання на нього) Certification Basis (Сертифікаційний базис)	Логотип EASA/FAA, багатостраничний технічний документ.

3. Ознаками високої складності ідентифікації є такі, які неможливо перевірити на скані, вони вимагають фізичного контакту з оригіналом документа. До таких ознак віднесемо печатки тиснені (рельєфні), водяні знаки, якість паперу чи відвідних бланків, справжність підпису (графологія).

2.2 Метод перевірки сертифікаційних авіаційних документів

Ключовою особливістю сертифікаційних документів є мультимодальність: вони містять комбінацію різних типів інформації, кожен з яких несе критичне значення для результатів перевірки усього документа. Властивість мультимодальності робить неможливим використання єдиного простого підходу. Наприклад, метод, який аналізує лише текст, пропустить підроблену печатку чи її відсутність, а метод, який аналізує лише візуальні елементи, пропустить невірні чи прострочені дати в тексті. Таким чином, доцільною виглядає розробка гібридного методу, який здатен працювати одночасно з усіма модальностями таких документів.

Для перевірки сертифікаційних авіаційних документів пропонується гібридний метод, який складається з наступних етапів:

1. Структурне моделювання документу (представлення цифрового вигляду документу у формальному структурованому описі).

2. Класифікація документу (віднесення його до певного виду сертифікаційного авіаційного документу, як то сертифікат експлуатації, сертифікат типу, тощо);

3. Верифікація документу.

4. Валідація документу.

5. Формування результатів перевірки документу.

Конвеєр автоматизованої перевірки документу у системі, що розробляється, відповідно до загального процесу автоматизованої перевірки документів (п.1) та з вказанням етапів розробленого методу наведено на рисунку 2.3.

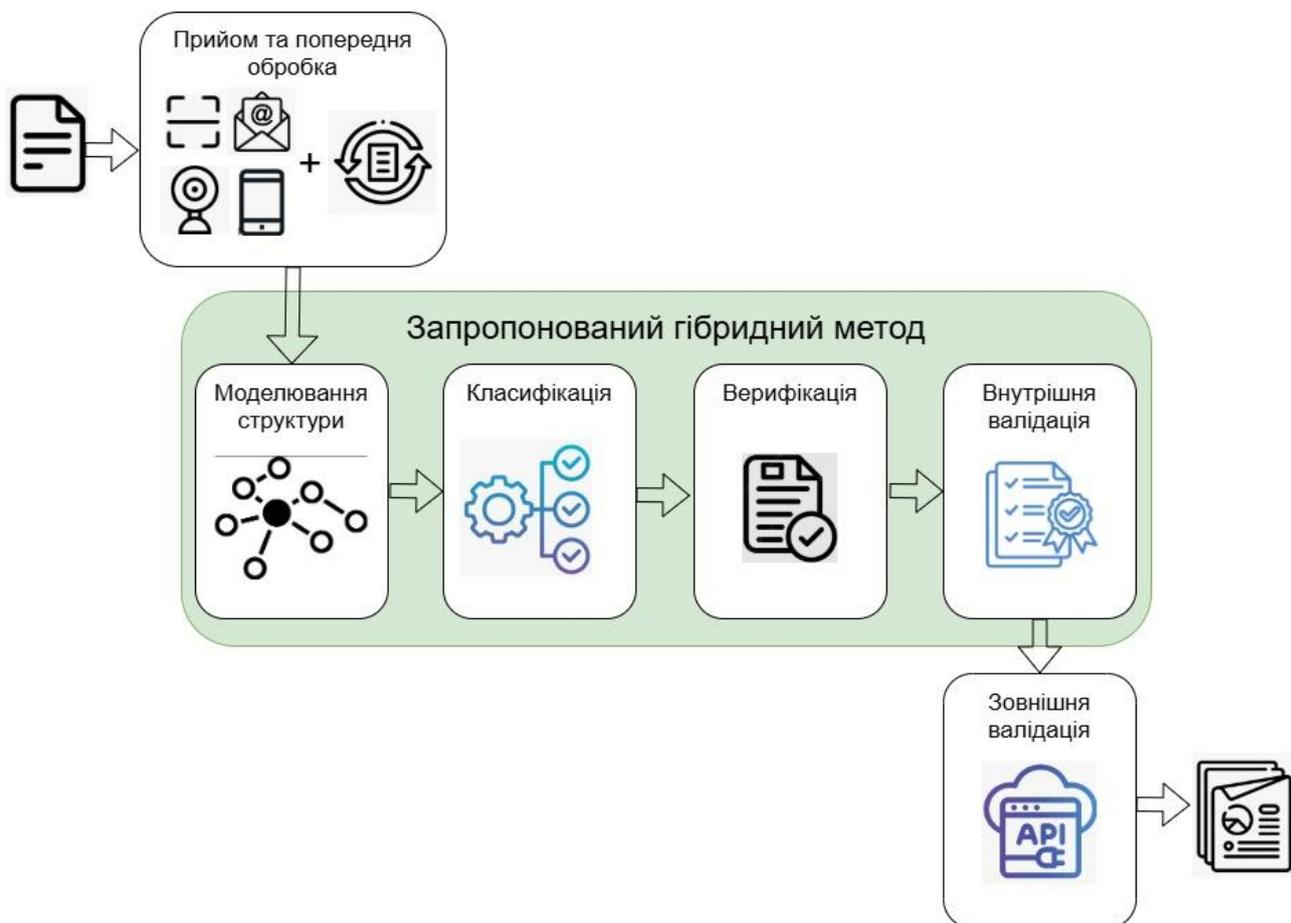


Рис. 2.3 Конвеєр автоматизованої перевірки сертифікаційних авіаційних документів

2.2.1 Математична модель задачі перевірки сертифікаційних авіаційних документів

Для визначення етапів та вибору засобів реалізації перевірки сертифікаційних авіаційних документів, побудуємо математичну модель задачі.

1. Формальна модель документа. Нехай сертифікаційний авіаційний документ описується множиною його окремих елементів $\mathcal{E}(d)$:

$$\mathcal{E}(d) = \{e_i\}_{i=1}^n, e_i = (b_i, t_i, c_i, s_i, v_i), \quad (2.1)$$

де: n – кількість елементів документа;

$b_i = (x, y_i, w_i, h_i)$ – координати області елемента;

$t_i \in T$ – тип елемента;

$c_i \in [0,1]$ – довіра до визначення елемента детектором;

s_i – текстовий елемент;

v_i – візуальний елемент.

2. Мультимодальний графовий простір. Для можливості подальшого аналізу документу, його формальний простір перетворимо на граф G_d , де вузли графа – елементи документу, ребра – просторові зв'язки:

$$G_d = (V_d, E_d), V_d = \{v_i | e_i \in \mathcal{E}(d)\}, |V_d| = n, \quad (2.2)$$

Ребра будуються на основі принципу k -найближчих сусідів:

$$E_d = \{(i, j) | j \in kNN(i)\}. \quad (2.3)$$

Вузли графа кодуються вектором ознак:

$$x_i = (p_i, c_i, o(t_i), em_i^{text}, em_i^{vis}), \quad (2.4)$$

де: p_i – координати та розміри;

c_i – довіра;

$o(t_i)$ – тип елемента;

em_i^{text} – embedding (кодування) текстового об'єкту;

em_i^{vis} – embedding (кодування) візуального об'єкту.

3. Векторизація графа G_d .

Врахування локального контексту вузлів (інформації про сусідів). На шарі l , стан вузла $h_i^{(l)}$ оновлюється на основі його попереднього стану та агрегованого стану його сусідів $\mathcal{N}(i)$:

$$h_i^{(0)} = x_i, c_i^{(l)} = \text{Context} \left(\{h_j^{(l-1)}\} : j \in \mathcal{N}(i) \right), \quad (2.5)$$

$$h_i^{(l)} = \text{Update}(h_i^{(l-1)}, c_i^{(l)}), h_i = h_i^{(L)}, \quad (2.6)$$

де: $h_i^{(0)}$ – початкова ознака вузла i ;

$\mathcal{N}(i)$ – множина сусідів вузла i ;

h_i - кінцева ознака вузла i .

Фінальний вектор документу g_d отримуємо кодуванням за допомогою глобального пулінга:

$$g_d = Pool(h_1, h_2, \dots, h_n). \quad (2.7)$$

3. Класифікація типу документа. Для визначення класу документа (наприклад, АОС, Part-145 та інші) використовуємо класифікатор на основі вектору g_d :

$$z = F(g_d), p = \underset{k}{\operatorname{softmax}}(z), \hat{y} = \underset{k}{\operatorname{argmax}} p(c_k), \quad (2.8)$$

$$\hat{y} \in \{1, \dots, |C|\} C = \{AOC, CofA, Part145, Part21G, TC\}, \quad (2.9)$$

де: p – розподіл ймовірностей між класами;

\hat{y} – прогнозний клас документу.

4. Гібридна модель верифікації.

Логічна верифікація (rule-based). Визначення обов'язкових полів прогнозованого типу документа:

$$u_d = \{(f, val_f) | f \in \mathcal{F}_{\hat{y}}\}, \quad (2.10)$$

де: $\mathcal{F}_{\hat{y}}$ – обов'язкові поля для типу документа \hat{y} ;

f – назва поля за шаблоном;

val_f – реальне значення з документа.

Визначення множини правил $R_{\hat{y}}$ для прогнозного типу документа:

$$R_{\hat{y}} = \{r_1, r_2, \dots, r_K\}, \quad (2.11)$$

де: \hat{y} – прогнозний клас документу.

Оцінка відповідності полів документу u_d за набором правил $R_{\hat{y}}$ для прогнозного типу документу.

$$S_d^{rules} = \frac{1}{K} \sum_{k=1}^K r_k(u_d), \quad (2.12)$$

де: $r_k \in R_{\hat{y}}$ – правило з множини правил;

$r_k(u_d) \in \{0,1\}$ – 1, якщо перевірка пройшла успішно, 0, якщо знайдено помилку.

Візуальна верифікація. Оцінка автентичності графічних елементів (логотипи, печатки, підписи). За енкодером автентичності отримуємо векторні ознаки z_i візуальних елементів v_i :

$$z_i = f_{auth}^{(\tau)}(v_i), \quad (2.13)$$

де: $v_i \in \mathcal{V}$ – візуальний елемент;

\mathcal{V} – множина візуальних елементів $\mathcal{V} = \{v_1, v_2, \dots, v_m\}$;

Агрегація візуальних ознак (механізм уваги):.

$$g_{vis} = \sum_{i=1}^m \alpha_i z_i, \quad \sum_{i=1}^m \alpha_i = 1, \quad (2.14)$$

де: α_i – вага, що визначає інформативність елемента.

5. Інтегральна оцінка дійсності документу та фінальний статус документу.

Інтегральна оцінка дійсності документу S_d обраховується як зважена сума довіри до класифікатора, оцінки виконання логічних правил та оцінки візуальної верифікації:

$$S_d = \lambda_1 conf_d^{cls} + \lambda_2 S_d^{rules} + \lambda_3 S_d^{graf}, \quad \lambda_i \geq 0, \quad \sum_i \lambda_i = 1 \quad (2.15)$$

де: $conf_d^{cls}$ – довіра до класифікатора;

S_d^{rules} – оцінка відповідності логічним правилам;

S_d^{graf} – графічна автентичність, частка елементів, що пройшли поріг «підозрілості» τ_g^{sus} :

$$S_d^{graf} = \frac{1}{|G|} \sum_{g \in G} I[S_g \geq \tau_g^{sus}], \quad (2.16)$$

де: s_g – метрична оцінка елемента типу g ;

τ_g^{sus} – поріг підозрілості.

Фінальний статус документа. Рішення, щодо фінального статусу документу визначається функцією, на основі порогів θ та критичних помилок δ :

$$FinalStatus(d) = \begin{cases} Valid, & S_d \geq \theta_P \cap \sum_k \delta_{d,k} = 0, \\ Partially\ valid, & S_d \in [\theta_P, \theta_V) \cap \sum_k \delta_{d,k} \leq \rho \\ Suspicious, & S_d \in [\theta_S, \theta_P) \cup \exists g: stuts_g = suspicious, \\ Invalid, & S_d < \theta_S \cup \exists k: \delta_{d,k} = 1(critical) \\ Forged, & \exists g: stutus_g = forged. \end{cases} \quad (2.17)$$

де: $\delta_{d,k} \in \{0,1\}$ - ознака критичної помилки правила k ;

$\theta_V, \theta_P, \theta_S$ – порогові значення серйозності;

ρ – допустима кількість помилок.

2.2.2 Класифікація сертифікаційних авіаційних документів

Найпростіший способом швидкої класифікації виглядає побудова вектору ознак вхідного документа за кількістю елементів: текстових блоків, логотипів, печаток тощо. Вектор подається на вхід алгоритму k -найближчих сусідів (k -NN), який миттєво порівнює вхідний вектор з векторами в базі шаблонів і знаходить найближчого сусіда, таким чином визначаючи клас документа, що перевіряється. Однак, такий або подібний підхід, має ряд критичних недоліків, як то наступні.

1. Векторизації макета на основі підрахунку елементів створює залежність від результатів роботи модулю аналізу макету. Якщо, сертифікат, що перевіряється, надходить має дуже низьку якість, через злиття тексту, або сильний візуальний шум модуль може помилково об'єднати кілька текстових елементів у один. Алгоритму, що оперує метриками відстані, може визначити цей макет ближчим до іншого шаблону.

2. Кількісний вектор ігнорує макет, у сенсі взаємовідносин елементів. Два документи з однаковим набором ознак алгоритм не зможе розрізнити. У той же час, документ з таким саме вектором ознак, може мати хаотичне розміщення ознак і не відповідати шаблону.

Таким чином, для реалізації надійної класифікації шаблонних документів, до яких належать сертифікаційні авіаційні документи метод повинен моделювати як просторові так і логічні взаємовідносини між елементами ознак. З цієї точки зору доцільним виглядає використання графових нейронних мереж GNN.

2.2.3 Структурне моделювання документу з використанням графових нейронних мереж

Графовий аналіз макета базується на представленні документа як графа, таким чином, що елементи макета, такі як текстові блоки та зображення, представлені як вузли, а різні просторові або логічні зв'язки між цими елементами представлені як ребра. Графові нейронні мережі GNN використовуються для прогнозування конкретних вузлів та підграфів як елементів макета, а також для прогнозування зв'язків між цими елементами через ребра в графах [55].

Перевагами використання таких моделей можна визначити наступні.

1. GNN моделюють саме відношення між вузлами. Інформація про те, що «логотип знаходиться над заголовком», явно кодується у структурі графа та використовується моделлю;

2. GNN моделі є надійними до помилок сегментації: втрата одного вузла або додавання «шумового» вузла не руйнує весь граф, а лише незначно змінює процес, що призводить до плавного зниження точності, а не до відмови у віднесенні документу до певного виду сертифікату [56].

3. GNN-моделі є відносно легкими, в порівнянні з моделями трансформерів, що робить їх ефективними для автоматизованих систем перевірки, де треба забезпечувати баланс між точністю та обчислювальними ресурсами системи [57].

2.2.4 Опис методу перевірки сертифікаційних авіаційних документів

Етап 1. Структурне моделювання документу

Вхідні дані етапу: документ у електронному вигляді.

Вихідні дані етапу: мультимодальний граф документа, векторизовані вузли графу, векторизований граф документу.

Крок 1. Аналіз макету. Передбачає виконання таких операцій над документом у електронному вигляді.

1. Виявлення елементів документа Для виконання операції використовується LayoutParser open-source бібліотека Python для аналізу макета документів. В результаті отримується список об'єктів типу LayoutElement, де кожен елемент містить координати обмежувальної рамки $[x_c, y_c, width, height]$, тип елемента та оцінку ймовірності, що елемент класифіковано правильно.
2. Парсинг текстових елементів. Для виконання операції використовується вбудований OCR-агент LayoutParser для Tesseract (open-source система оптичного розпізнавання тексту OCR), який нормалізує текст, виконує пошук слів-шаблонів, розпізнає структури тексту.

Приклад отриманого на цьому кроці результату у спрощеному вигляді за структурою JSON наведено на рисунку 2.4.

```
[
  {
    "type": "Logo",
    "bbox": [20, 20, 140, 90],
    "text": "",
    "field": "organization_logo",
    "confidence": 0.97
  },
  {
    "type": "Title",
    "bbox": [160, 20, 580, 110],
    "text": "СЕРТИФІКАТ ВІДПОВІДНОСТІ",
    "confidence": 0.99
  },
  {
    "type": "Text",
    "bbox": [20, 130, 580, 180],
    "text": "Номер: UA.12345.2025",
    "normalized": "UA-12345-2025",
    "field": "certificate_number",
    "confidence": 0.95
  },
  {
    "type": "Text",
    "bbox": [20, 190, 580, 240],
    "text": "Дата видачі: 12.04.2025",
    "normalized": "2025-04-12",
    "field": "issue_date",
    "confidence": 0.94
  },
  {
    "type": "Stamp",
    "bbox": [30, 500, 200, 650],
    "text": "",
    "field": "official_stamp",
    "confidence": 0.93
  },
  {
    "type": "Signature",
    "bbox": [350, 500, 580, 650],
    "text": "",
    "field": "signature_area",
    "confidence": 0.88
  }
]
```

Рис 2.4 Результат аналізу макету у вигляді структури JSON

Крок 2. Побудова мультимодального графу.

Модель документа у форматі JSON, що було побудовано Кроці 1 перетворюється в графову модель наступним чином.

1. Вузли графу формуються з елементів списку макета, що було виявлено на Кроку 1.

2. Ознаки вузлів графу. Оскільки нейронні мережі можуть працювати лише з числами, кожен вузол повинен мати вектор ознак. Для моделі, що використовується, вектор ознак елемента представлено конкатенацією наступних компонент:

- просторові ознаки – описують геометричні характеристики елемента, уявляють собою нормалізовані відносно розміру документа координати центру елемента і його обмежувальної рамки $[x_c, y_c, width, height]$;
- ймовірність правильного визначення елемента (впевненість, довіра, або confidence);
- тип елемента у вигляді вектору з нулів та однієї одиниці, довжиною k , де k – кількість типів елементів, яка стоїть на позиції, що відповідає типу елемента – вектор «one-hot type»;
- єдиний мультимодальний семантичний embedding – цифрове кодування, що об'єднує два компоненти для текстових та графічних блоків (якщо елемент не містить тексту, текстовий компонент заповнюється нулями, якщо елемент не містить графічної інформації, нулями заповнюється візуальний компонент), довжина якого залежить від моделей кодування. Структуру вектору ознак вузла наведено на рисунку 2.2.

```
[
  cx, cy, w, h,           # геометричні ознаки
  confidence,           # ймовірність правильної класифікації
  one_hot(type),        # вектор one-hot type елемента
  semantic_embedding (384 + 512) # єдиний embedding
]
```

Рис.2.4 Структура вектору ознак вузла графу

3. Ребра графу уявляють собою просторові взаємозв'язки між вершинами графу (елементами документу) та моделюються за допомогою підходу k -closest-neighbor (k NN), коли кожна вершина з'єднана лише з k вершинами, що є її найближчими сусідами за евклідовою відстанню між координатами центру елемента. Це дозволить вивчати локальні структури макету документу та не перевантажувати структурний граф просторовими відношеннями «усіх до усіх» без втрати інформації, щодо структури документу та взаємного розташування його елементів. На рисунку 2.5 наведено приклад побудови такого графа для $k = 2$ на умовному макеті сертифікаційного документа з узагальненими блоками.

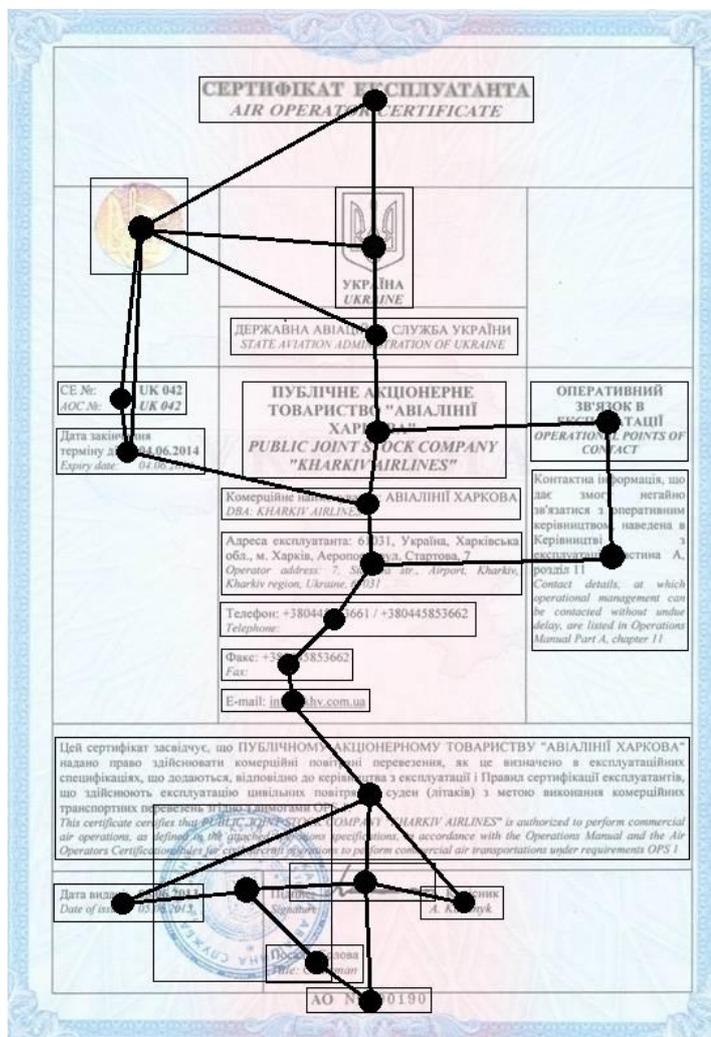


Рис.2.5 Приклад побудови k -closest-neighbor графа для $k = 2$

Крок 4. Векторизація графу.

Для формування єдиного представлення всього документу будемо використовувати метод глобальної агрегації (global mean pooling) нейронної мережі GraphSAGE.

1. Агрегація повідомлень. Модель GNN виконує L шарів агрегації повідомлень. На кожному рівні l вектор-представлення вузла $h_v^{(l)}$ оновлюється на основі векторів його сусідів $h_v^{(l-1)}$, саме це забезпечує поширення структурної та семантичної інформації по графу.

2. Глобальний пулінг. Після завершення L шарів отримані вектори вузлів $\{h_v^{(L)}\}$ агрегуються за допомогою операції глобального пулінгу. В результаті отримуємо єдиний вектор-представлення графа h_G , - узагальнене представлення документа.

Етап 2. Класифікація документа

Вхідні дані: мультимодальний граф документа, векторизовані вузли графу, векторизований граф документу, отримані на Етапі 1.

Вихідні дані: прогнозований тип документу, ймовірність, що документ класифіковано правильно (confidence), попередній висновок про наявність структурних аномалій.

Для отримання прогнозу, щодо типу документу будемо використовувати багатошаровий перцептрон (MLP), на вхід якого буде подаватися векторизоване уявлення графу документу та фінальну функцію Softmax, , яка здійснює віднесення документа до певного класу: «Сертифікат експлуатанта», «Сертифікат льотної придатності» і т.д.

Етап 3. Верифікація документа

Вхідні дані: нормалізовані текстові поля, векторні уявлення текстових та графічних елементів, регулярні вирази та шаблони номерів (для кожного типу документів), спеціалізовані довідники, довідники унікальних структурних та змістових полів (згідно таблиці 2.1).

Вихідні дані: перелік коректних/некоректних полів документу, перелік знайдених помилок, оцінка рівня відповідності типу документу, ознаки некоректності полів:

- недійсний номер;
- неправильна дата;
- відсутнє унікальне поле для відповідного типу;
- аномальний логотип, печатка, підпис.

Мета етапу перевірити коректність і логічну узгодженість змістових полів документа. Тут виконується смислова перевірка змісту виявлених текстових та числових полів на правильність та коректність значень, відповідність форматам та логічних взаємозв'язків. В процесі верифікації проводиться аналіз значення текстових полів, форматів номерів, достовірність дат, відповідність регуляторним вимогам, а також перевірка специфічних ознак, які характерні для певних типів сертифікатів. Для проведення верифікації було визначено наступні перевірки.

1. Формат номера сертифіката – відповідно до кожного типу сертифікаційного авіаційного документу застосовується регулярний вираз, що відповідає формату номеру документа (табл. 2.2).

2. Перевірка формату дат:

- перевірка форматів: DD.MM.YYYY, YYYY-MM-DD, DD/MM/YYYY;
- коректність чисел (місяці 1–12, дні 1–31);
- перетворення у уніфікований формат ISO 8601.

3. Часова логічна перевірка дат: дата видачі \leq поточна дата та дата закінчення \geq дата видачі.

4. Перевірка наявності всіх дат:

- дата видачі;
- дата набрання чинності;
- дата закінчення терміну дії.

5. Виявлення помилок порівнянням векторів «ключових слів» із заголовку документу (згідно таблиці 2.1).

6. Перевірка наявності та повноти обов'язкових полів для кожного типу документу (згідно таблиці 2.1).

7. Перевірка графічних елементів:

- наявність логотипу EASA / національного органу;
- перевірка форми та розташування печатки.

Таблиця 2.2

Регулярні вирази перевірки форматів номерів сертифікатів

Тип документа	Формат номеру	Приклад	Регулярний вираз
Сертифікат експлуатанта (АОС)	[Код країни].АОС.XXX	UA.АОС.001	$^{[A-Z]{2}}\backslash\text{.АОС}\backslash\backslash\text{d}\{3\}\$$
Схвалення організації з ТО (Part-145)	[Орган].145.XXXX	EASA.145.1234	$^{[A-Z]}\backslash\backslash\text{.145}\backslash\backslash\text{d}\{4\}\$$
Схвалення організації з виробництва (Part-21G)	[Орган].21G.XXXX	EASA.21G.1234	$^{[A-Z]}\backslash\backslash\text{.21G}\backslash\backslash\text{d}\{4\}\$$
Сертифікат типу (ТС)	[Орган].[Категорія].XXX	EASA.A.064	$^{[A-Z]}\backslash\backslash\text{.}[A-Z]\backslash\backslash\text{d}\{3\}\$$
Сертифікат льотної придатності (CofA)	нестандартизований		залежить від держави

8. Перевірка логічної узгодженості документу:

- відповідність між типом документа і полями;
- відповідність між полями модель повітряного судна - серійний номер - дані сертифіката типу;
- відповідність номера типу документа (згідно таблиці 2.1);
- відповідність між номерами сертифікатів у кількох місцях документа.

Етап 4. Внутрішня валідація документу

Вхідні дані: прогнозований тип документа з Етапу 2, результати верифікації з Етапу 3, векторне представлення графічних елементів (підпис, печатка, логотип).

Мета етапу визначити дійсність та автентичність документа на основі графічних та текстових ознак, а також відповідності регуляторним вимогам.

Крок 1. Мультиmodalне злиття ознак графічних елементів

Embedding документа, отриманий в Етапі 1 є структурним, тобто відображає макет документа, він не містить достатньої інформації для оцінки автентичності графічних елементів сертифікаційного документа (печатка, підпис, логотип). На даному кроці будемо формувати графічний embedding, шляхом attention-based feature fusion (механізм уваги), що узагальнить локальні графічні ознаки документа та підвищить стійкість методу перевірки до шумів та часткових пошкоджень документа. Вектори графічних елементів будемо пропускати через модуль уваги (1-шаровий MLP), який вивчає динамічні ваги:

- w_{logo} - інформативність ознак логотипу;
- w_{stamp} - інформативність ознак печатки;
- $w_{signature}$ - інформативність ознак підпису.

Фінальний вектор ознак формуємо як зважену суму:

$$h_{final} = (w_{logo} \cdot h_{logo}) + (w_{stamp} \cdot h_{stamp}) + (w_{signature} \cdot h_{signature}), \quad (2.18)$$

де: $w_{logo} + w_{stamp} + w_{signature} = 1$.

Модуль уваги поступово знижує вагу неінформативних ознак. Якщо мережа бачить, що вектор h_{stamp} є неінформативним (може бути у випадку змазаного зображення), вона навчиться присвоювати йому низьку вагу ($w_{stamp} = 0.1$) і покладатися переважно на $h_{signature}$ ($w_{signature} = 0.7$).

Наприклад, при змазаній печатці мережа знижує її вагу до $w_{stamp} = 0.1$, водночас збільшуючи роль підпису $w_{signature} = 0.7$. Це забезпечує методу перевірки адаптивність та стійкість до реальних умов сканування та фотозйомки.

Побудована таким чином векторна ознака буде використовуватися на кроці 3 для оцінки справжності, цілісності та відповідності документа.

Крок 2. Верифікація автентичності графічних елементів

Для задачі верифікації графічних елементів (логотипу, печатки та підпису) будемо використовувати нейронну архітектуру Triplet Network, яка формує латентний простір, у якому зразки автентичних підписів певної особи розташовуються близько один до одного, тоді як зразки інших осіб – на значній відстані [58].

Мережа навчається на наборах трійок: Anchor – зразок, що перевіряється, Positive – зразок оригінал, Negative - зразок іншого елементу або відома підробка. Функція втрат Triplet Loss навчить мережу, щоб відстань між Anchor та Positive плюс певний зазор була меншою за відстань між Anchor та Negative:

$$Loss = \max(\|h_A - h_P\|^2 - \|h_A - h_N\|^2 + margin, 0) \quad (2.19)$$

Крок 3. Валідація на основі правил

Для проведення верифікації було визначено наступні перевірки.

1. Перевірка допустимої тривалості дії

- "АОС чинний до внесення змін"
- "CofA раз на 12 місяців"
- "21G діє до скасування"

2. Перевірка дійсності графічних елементів

- відповідність печатки еталонному шаблону
- відповідність підпису reference-підпису
- оцінка схожості \geq threshold

3. Регуляторні правила (від EASA/FAA)

- якщо CofA без Registration Marks, то документ недійсний
- якщо Part-145 без Rating категорій, то Non-compliant

4. Перевірка допустимої тривалості дії

- АОС є чинним до внесення змін;
- CofA є чинним упродовж на 12 місяців;
- Maintenance Approval є чинним безстроково, але з умовами;
- Part-21G є чинним, допоки не буде скасовано регулятором.

На цьому кроці здійснюється комплексна оцінка дійсності документа на основі результатів класифікації та верифікації. Для цього будемо застосовувати набір правил і логічних умов, які дозволять встановити, чи відповідає документ вимогам регулятора і чи не містить він ознак підробки.

Приклад системи правил валідації.

1. Правила обов'язкових полів

```
IF missing(certificate_number) → document = Invalid
IF missing(issue_date) → document = Invalid
IF missing(stamp OR signature) → document = Suspicious
```

2. Правила часової відповідності

```
IF issue_date > current_date → Invalid
IF expiry_date < current_date → Expired
```

3. Правила регуляторної відповідності

```
IF document_type == CofA AND missing(Aircraft Manufacturer) → Non-compliant
IF certificate_number NOT in known_range → Suspicious
```

4. Правила узгодженості структури

```
IF title_position far from typical region → Wrong Layout
IF stamp too small / distorted → Potentially forged
```

5. Правила аномалій графічних компонентів

```
IF CLIP similarity(stamp, reference_stamp) < threshold → Suspicious stamp
IF signature embedding length < threshold → Low-quality signature
```

Вихідні дані:

1. Оцінка автентичності графічних елементів:

- статус підпису: valid / suspicious / forged;
- статус печатки: valid / inconsistent / forged;
- статус логотипу: authentic / mismatched;

2. Аналіз відповідності регуляторним вимогам:

- термін дії документа (current / expired / indefinite);
- відповідність типу сертифіката структурним ознакам;
- відповідність унікальних полів вимогам типу (згідно табл. 2.1).

3. Інтегральна оцінка дійсності документа;

4. Перелік порушень правил;

5. Фінальний статус документа:

- Valid (дійсний);
- Partially valid (частково дійсний, містить зауваження);
- Invalid (недійсний);
- Suspicious (ознаки фальсифікації);
- Forged (висока ймовірність підробки).

Узагальнений алгоритм запропонованого гібридного методу наведено на рисунку 2.6.

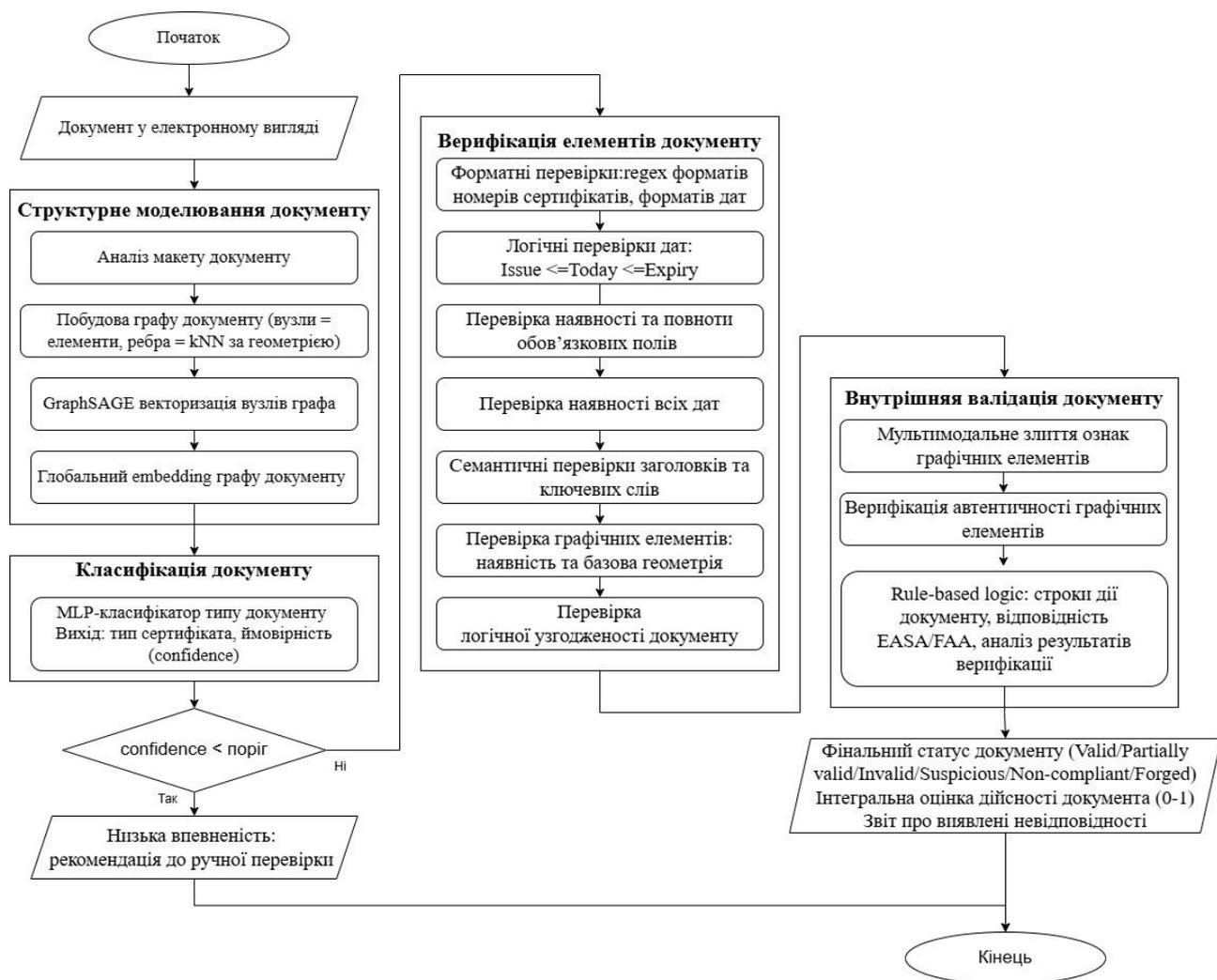


Рис.2.6 Узагальнений алгоритм запропонованого гібридного методу перевірки сертифікаційних документів

2.3 Автоматизована система перевірки авіаційних документів

Актуальність автоматизації процесу перевірки сертифікаційних авіаційних документів зумовлена великим обсягом документації, низькою швидкістю ручної перевірки та низькою ефективністю процесу аудиту та формування звітності.

До автоматизованої системи перевірки документів висуваються такі основні функціональні вимоги: отримання цифрового уявлення документу, його зберігання та обробка (перевірка валідності, автентифікація та винесення рішення щодо його дійсності) [59].

Для автоматизованої системи перевірки авіаційних документів пропонується клієнт-серверна архітектура. Вибір архітектури ґрунтується на необхідності ефективного масштабування обчислювальних потужностей для обробки великих обсягів документів та забезпечення надійного доступу до системи для авторизованих користувачів незалежно від їхнього географічного розташування через мережу Інтернет. Уся «важка» робота системи, як то обробка документів, верифікація графічних елементів, перевірка баз даних відбувається на стороні сервера. При збільшенні об'ємів документів, що перевіряються, нарощується потужність сервера, не чіпаючи пристрої користувачів. Користувачам (інспекторам, персоналу аеропорту, іншим службам) для перевірки сертифікату потрібен лише клієнт системи, яким може виступати простий веб-браузер, легкий мобільний додаток або доступ до API. Вони надсилають запит на сервер «перевір цей документ» і отримують відповідь «ОК» або «Підробка» з будь-якої точки світу, де є Інтернет. Архітектуру «клієнт – сервер» наведено на рисунку 2.7.

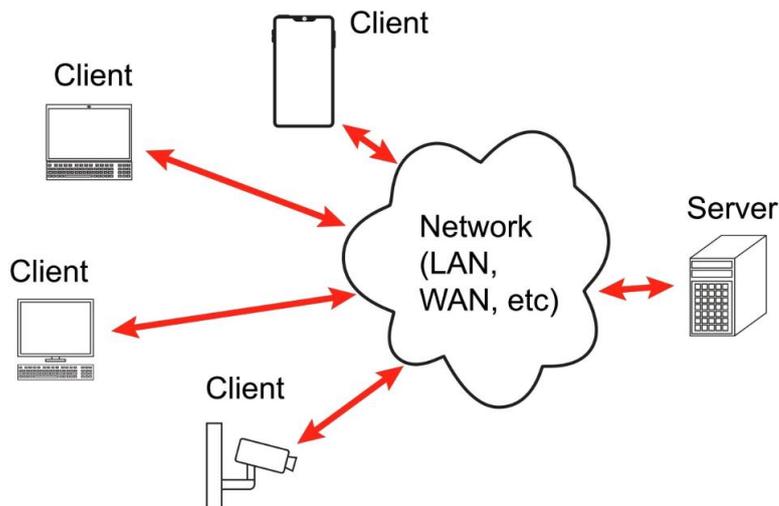


Рис. 2.7 Архітектура «клієнт-сервер»

Беручи до уваги специфіку авіаційної галузі та сертифікаційних авіаційних документів, окрім масштабування та віддаленого доступу, клієнт-серверна архітектура надає наступні важливі переваги.

1. Централізована безпека та контроль доступу. Уся «логіка» верифікації документу (навчена ШІ-модель) та бази даних (зразки справжніх печаток, підписів, сертифікати-зразки) знаходяться на захищеному сервері, що забезпечує:

- захист моделі – модель перевірки сертифікаційних документів є унікальною, уявляє собою інтелектуальну власність та ключ до безпеки;
- керований доступ – доступ до даних та моделей системи є чітко контрольованим та обмеженим набором прав для відповідних груп користувачів.

2. Швидке оновлення та зручне обслуговування системи. У випадках, коли з'являється новий тип підробки, приймається новий зразок сертифікаційних документів або інша подібна ситуація оновлення даних, оновлення моделі перевірки, шаблону документу або логіки надання даних виконується лише в одному місці на сервері. Відразу після оновлення усі користувачі системи (клієнти) будь де у будь якій географічній точці мають можливість користуватися новою,

оновленою логікою. У цьому випадку, немає потреби оновлювати програмне забезпечення на кожному клієнтському пристрої, що є критичним для швидкої реакції на загрози.

3. Цілісність та узгодженість даних. Усі клієнти звертаються запитом до єдиного джерела даних: центральної бази даних та моделі перевірки документів. Незалежно від точки доступу, клієнти, що перевіряють один і той самий документ, отримують однакову актуальну відповідь, що виключає ситуації використання локальних копій баз даних та моделей перевірки і проходження перевірки недійсним документом.

4. Ефективність ресурсів та «тонкі клієнти». Модель перевірки документу з використанням штучного інтелекту є вимогливою до обчислювальних ресурсів. Пристрій користувача виконує прості та легкі задачі отримання зображення документу, відправка документу та запиту на його перевірку до сервера та отримання відповіді. За рахунок використання клієнт-серверної технології досягаються наступні цілі:

- економія грошових витрат на придбання дорогих та потужних пристроїв для кожного клієнта;
- швидкість отримання відповіді при обробці запиту на спеціалізованому серверному процесорі значно вища, за швидкість перевірки на мобільному процесорі.

Використання штучного інтелекту в процесі автоматизованої перевірки вимагають значних обчислювальних ресурсів, які недоцільно розміщувати на клієнтських пристроях:

- серверна частина системи виконує всі складні для обчислювання задачі: попередню обробку зображень, виконання моделей перевірки, зберігання даних (еталонні вектори, шаблони) та централізоване управління процесами перевірки;
- клієнтська частина (веб-інтерфейс або API для інтеграції з іншими системами) надсилає запити на перевірку у вигляді зображення документів та отримує готові результати у вигляді звітів.

Для автоматизованої системи перевірки сертифікаційних авіаційних документів пропонується структура, яку наведено на рисунку 2.8.

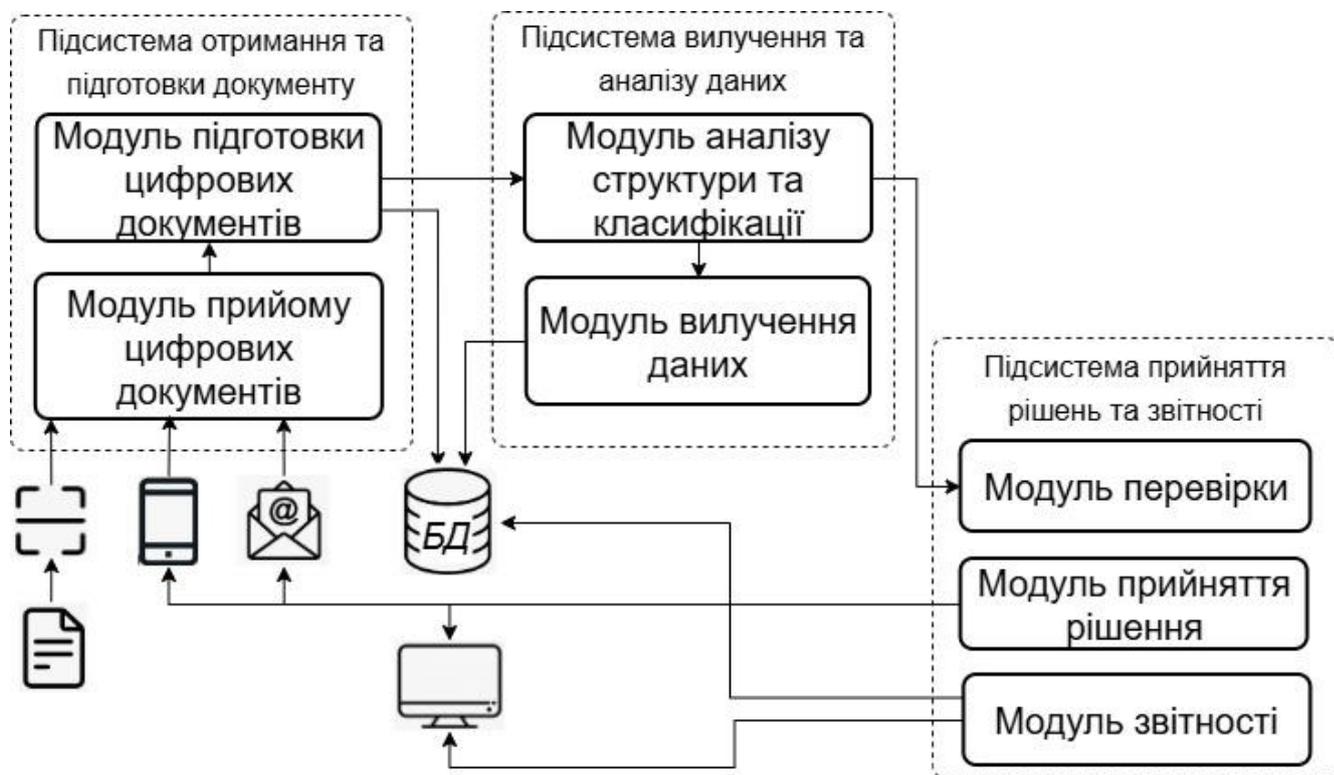


Рис. 2.8 Структура автоматизованої системи перевірки авіаційних документів

Структура автоматизованої системи перевірки авіаційних документів складається з наступних підсистем та модулів [59]:

1) підсистема отримання та підготовки документу:

- модуль прийому цифрових документів – отримує цифрову версію документу (скан, фото, pdf файл) отримані по локальній мережі, мережі Інтернет;
- модуль підготовки цифрових документів – виконує обробку зображень з метою підвищення якості візуальних елементів;

2) підсистема вилучення та аналізу даних:

- модуль аналізу структури та класифікації – виконує аналіз макету, структури документу та його класифікацію;

- модуль вилучення даних – визначає типи елементів документу, та їх атрибути, вилучає та нормалізує дані (наприклад, приводить дати до єдиного формату), вилучає візуальні елементи (печатки, підписи, логотипи, тощо) для подальшої верифікації.

3) підсистема прийняття рішень та звітності:

- модуль перевірки – виконує верифікацію (наприклад, чи коректні дати, чи відповідають дані стандартам, тощо) та валідацію документу (чи є справжніми печатки, підписи, голограми, тощо), виявляє ознаки шахрайства та аналіз аномалій у документі;
- модуль прийняття рішень – підсумовує результати усіх перевірок та відповідно до обраного набору рішень приймає остаточне рішення, щодо результатів перевірки документу;
- модуль звітності – формує для перевіряючого результати перевірки документу у вигляді докладних звітів результати перевірки документу (з вказанням результатів кожного виду перевірки, підставами для прийняття остаточного рішення та можливими виявленими аномаліями чи невідповідностями) .

2.4 Програмна реалізація методу перевірки сертифікаційних авіаційних документів

2.4.1 Структура програмного проекту

Для реалізації запропонованого методу перевірки сертифікаційних авіаційних документів було розроблено програмне забезпечення. Середовищем розробки було обрано MS Visual Studio 2022. Вибір інтегрованого середовища ґрунтувався на наявності широкого інструментарію розробки .NET додатків та можливості реалізації багатопроєктних додатків [60]. Для реалізації було обрано багат шарову архітектуру, перевагами використання якої є наступні:

- чіткий розподіл інтерфейсу користувача, бізнес-логіки, моделі доменів та інтеграцій;

- можливість модернізації або заміни цілого шару без впливу на інші шари;
- підвищена тестованість – можливість тестувати шари без бази даних та веб-інтерфейсу.

Технологічною основою розробки користувацького інтерфейсу було обрано архітектурний шаблон Model–View–Controller, що було обумовлено наступними перевагами [61]:

- шаблон підходить до складних систем з багатьма модулями;
- забезпечує чіткий поділ поміж контролерами, подань та моделей;
- легке поєднання з зовнішніми сервісами та WebAPI.

Програмне рішення системи автоматизованої перевірки має назву *AviaCertVerification*, структуру рішення наведено на рисунку 2.9.

Рішення складається з шістьох проєктів. Розглянемо їх призначення.

AviaCertVerification.Application (class library project). Реалізує бізнес-логіку методу перевірки документів:

- прийом та підготовка документу;
- структурне моделювання: визначення текстових та візуальних елементів, побудова графа макету документів;
- класифікація документів та їх вузлів;
- верифікація текстових та візуальних елементів;
- внутрішня валідація документів та прийняття фінального рішення, щодо їх результату перевірки.

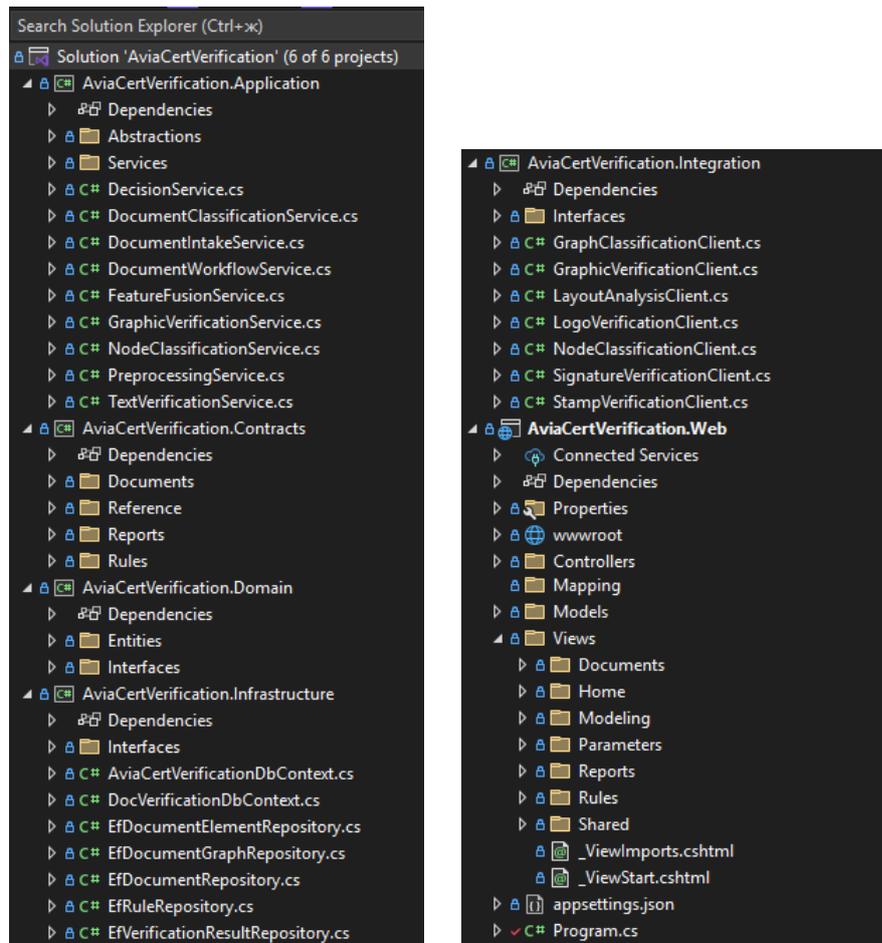


Рис. 2.9 Структура програмного рішення *AviaCertVerification*

Містить сервіси, що відповідають реалізованим бізнес-процесам: *DocumentWorkflowService*, *PreprocessingService*, *DocumentClassificationService*, *DecisionService* та ін.

Робота шару здійснюється через взаємодію з інтерфейсами *Domain* та клієнтами *Integration*. Виділення «ядра» бізнес-логіки в окремий проект дало можливість реалізувати алгоритмічну частину пропонованого методу без інтерфейсу та інфраструктури.

AviaCertVerification.Contracts (class library project). Відповідає за взаємну передачу структурованих даних між шарами. Містить прості класи, структури, які відображають конкретні об'єкти обміну інформацією між різними шарами рішення. Вони визначають які саме дані та у якому форматі передаються між шарами. Реалізовані структури забезпечують:

- передачу даних між *Web* та *Application*;

- передачу даних між Application та Integration;
- формування запитів/відповідей до Python сервісів;
- серіалізацію/десеріалізацію JSON.

Виділення контрактів у окремий проект забезпечує відокремлення бізнес-процесів від механізму обміну даними.

AviaCertVerification.Domain (class library project). Шар поєднує бізнес-сутності та правила предметної області та забезпечує їх незалежність від конкретних технологій, що користуються ними. Шар містить описи класів об'єктів предметної області, а також допоміжні структури до них:

1) класи: Document, DocumentElement, DocumentGraph, VerificationResult, RuleResult;

2) переліки: переліки (DocumentType, ElementType)

Також шар містить інтерфейси (визначення, яку поведінку мають реалізовувати класи), що визначають, як саме інші шари працюють з даними та сервісами: IDocumentRepository, IDocumentElementRepository, IDocumentGraphRepository, IverificationResultRepository та ін.

AviaCertVerification.Infrastructure (class library project). Забезпечує доступ до даних та внутрішньої інфраструктури. Зберігає внутрішні структури моделей документу, результатів класифікації та верифікації, містить реалізацію домених інтерфейсів, реалізує роботу з базою даних та іншими внутрішніми ресурсами: AviaCertVerificationDbContext, EfDocumentRepository, EfDocumentElementRepository, EfDocumentGraphRepository, EfVerificationRepository, EfRuleRepository та ін.

Винесення забезпечення доступу до даних у окремий проект дозволило відокремити бізнес-логіку та конкретних технічних моментів зберігання даних. Це дозволяє змінити базу даних, одну СУБД на іншу без зміни програмних сутностей.

AviaCertVerification.Integration (class library project). Шар інтеграції з зовнішніми сервісами, зокрема Python-сервером з моделями (LayoutParser, GNN, PyTorch Geometric, MLP Triplet Network). Реалізує інтерфейси: ILayoutAnalysisClient, IGraphClassificationClient, ILayoutAnalysisClient,

`IGraphClassificationClient`, `ILogoVerificationClient`, `ISignatureVerificationClient`, `IStampVerificationClient` та інші. Шар ізолює зовнішні залежності, що дозволяє просту та швидко заміну модулів та сервісів, а також включення/відключення реальної інтеграції.

AviaCertVerification.Web (ASP.NET Core MVC project). Презентаційний шар, який призначено для взаємодії з користувачем (завантаження документів, перегляд результатів перевірки, звітності). Відповідно до моделі MVC шар реалізує:

- моделі – класи для передачі даних контролерам (`DocumentViewModels`, `ModelingViewModel`, `ParametersViewModels`, `ReportViewModels`);
- контролери – відповідають за обробку запитів користувачів (`DocumentsController.cs`, `ReportsController`, `ParametersController`, тощо),
- представлення – відповідають за візуальне відображення даних користувачам (відображення документа та його структури, результатів класифікації та верифікації та керування процесом перевірки).

2.4.2 Інтерфейс користувача

Розроблене програмна система *AviaCertVerification* реалізує запропонований метод автоматизованої перевірки сертифікаційних авіаційних документів та демонструє його практичне застосування. Взаємодія системи з користувачем виконується за допомогою веб-інтерфейсу, що складається з декількох веб-сторінок. Головна сторінка містить короткий опис призначення системи та навігаційне меню, що розміщено у верхній її частині. Навігаційне меню у свою чергу надає доступ до інших веб-сторінок та включає такі пункти:

- 1) «Головна»;
- 2) «Документи»;
 - «Список документів»;
 - «Завантажити документ»;
- 3) «Моделювання»;
- 4) «Параметри»;
- 5) «Адміністрування».

Сторінку реалізовано Home/Index.cshtml.

Сторінка «Список документів» відображає перелік оброблених документів з вказанням назви документу (назва файлу), тип документу, статус та дату його перевірки. Сторінка дозволяє виконати фільтрацію переліку за типом документа, джерелом, або статусом, а також відкрити для перегляду певний документ.

Сторінка «Завантажити документ» пропонує користувачу обрати файл документу для завантаження, виконати його попередній перегляд та виконати автоматичну перевірку. Після перевірки на сторінці відображається панель з її результатами: тип документа (виявлений та очікуваний), статус перевірки, значення метрик оцінювання результатів перевірки, кнопки «Підтвердити», «Відхилити», «Перепризначити тип». Ця сторінка підтримує надання оцінки документа вручну. Сторінку реалізовано MVC: Models/DocumentViewModel.cs, Views/Documents/Upload.cshtml, Controllers/DocumentController.cs, зовнішній вигляд сторінки наведено на рис. 2.10.

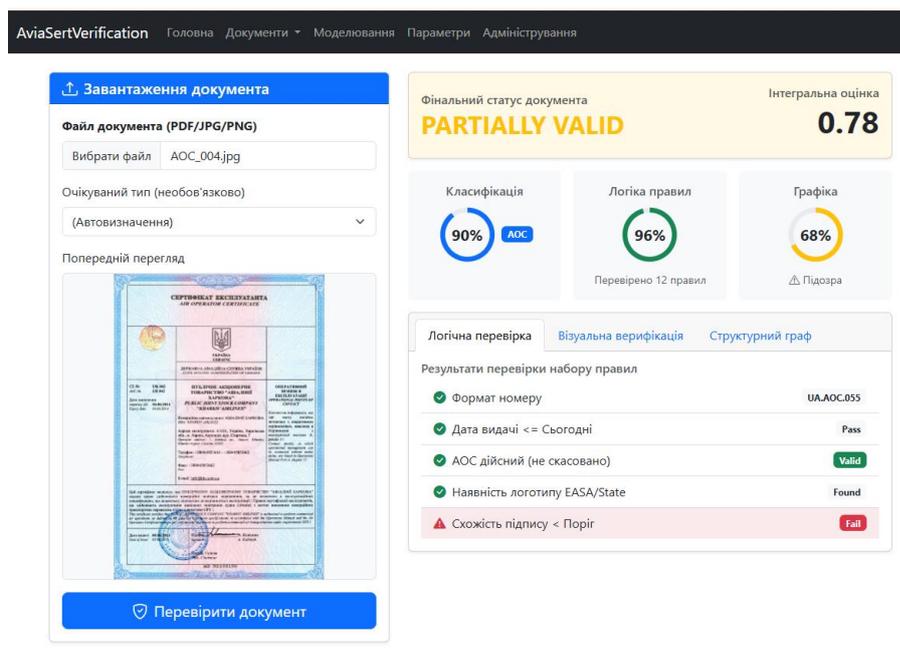
Сторінка «Моделювання» призначена для тестування перевірки групи документів. Ліва панель містить форму, де вказуються такі параметри: вибір директорії розміщення файлів документів, кількість документів, список типів документів для фільтрації та вибір режиму перевірки – повна(валідація) чи часткова (тільки класифікація). По натисканню кнопки «Запустити» запускається процес автоматизованої перевірки файлів документів [вказано] кількості з вказаної директорії. Права частина сторінки відображає панель результатів перевірки. Зведена інформацію про результати моделювання наводиться у вигляді дашборду: загальна кількість оброблених документів, кількості успішно підтверджених(Valid), частково валідних (Partially Valid), підозрілих(Suspicious), недійсних(Invalid) та підроблених (Forged). Нижче розміщено блок метрик якості перевірки (точність класифікації, точність валідації документів, точність верифікації текстових блоків, точність валідації графічних блоків, частка невиявлених недійсних документів та середній час перевірки). Під блоком метрик розміщено таблицю результатів по кожному документу: реальний та

прогнозований тип, статус перевірки, фінальна оцінка перевірки та час обробки. Сторінку реалізовано MVC:

Models/ModelingViewModel.cs,
Views/Modeling/Index.cshtml,
Controllers/ModelingController.cs,

зовнішній вигляд сторінки наведено на рисунку 2.11.

Сторінка «Параметри перевірки» призначена для налаштування основних параметрів методу перевірки документів. У верхній частині сторінки відображаються заголовок та інформаційне повідомлення, про вдале (чи невдале) збереження параметрів. Основний блок налаштування параметрів складається з секцій логічно згрупованих елементів, виявлення елементів, параметри структури графу, налаштування нейронної архітектури графу та його обробки, кодування графу на поріг довіри при класифікації, параметри перевірки текстових елементів, параметри перевірки графічних елементів, параметри інтегральної оцінки та прийняття фінального рішення. Кожен блок може бути згорнуто та розгорнуто, що є зручним для користувача при великій кількості параметрів. Сторінку реалізовано MVC: Models/ParametersViewModel.cs, Views/Parameters/Index.cshtml, Controllers/Parameters Controller.cs, зовнішній вигляд сторінки наведено на рисунку 2.12.



AviaCertVerification © 2025

Рис. 2.10 Сторінка завантаження та перевірки документа

AviaSertVerification Головна Документи ▾ Моделювання Параметри Адміністрування

Параметри тестування

Розташування документів
C:\My\AviationDocuments\Certificates\Set

Тестовий набір

Кількість документів
40

Тип документів (фільтрація)
(будь-який) ▾

Режим перевірки
 Часткова (класифікація)
 Повна

Запустити

Усього	Valid	PartiallyValid	Suspicious	Invalid	Forged
40	22	3	6	7	2

Метрики якості перевірки

Точність класифікації документа 95,0%	Точність класифікації текстових 89,0%
Точність валідації документа 90,0%	Точність верифікації візуальних 86,0%
Частка не виявлених недійсних 11,0%	Середній час обробки (сек)

Результати по документах

Файл	Тип (pred)	Статус
AOC_004.jpg	AOC	Invalid
CofA_003.jpg	CofA	Valid
Part145_002.jpg	Part145	Valid
TC_004.jpg	TC	Invalid
AOC_011.jpg	AOC	Partially Valid

Рис. 2.11 Сторінка моделювання перевірки документів

AviaSertVerification Головна Документи ▾ Моделювання Параметри Адміністрування

Параметри перевірки

- 1) Виявлення елементів ▾
- 2) Структура графа ▾
- 3) Параметри GNN (GraphSAGE) ▾
- 4) Embedding та класифікація типу ▾
- 5) Перевірка текстових елементів ▾
- 6) Перевірка графічних елементів ▾
- 7) Інтегральна оцінка і фінальний статус документа ▾

Вага класифікатора (λ_1) <input type="text" value="0,4"/>	Вага правил (λ_2) <input type="text" value="0,35"/>	Вага графічної автентичності (λ_3) <input type="text" value="0,25"/>
Попір valid (θ_v) <input type="text" value="0,85"/>	Попір partially valid (θ_p) <input type="text" value="0,7"/>	Попір suspicious (θ_s) <input type="text" value="0,5"/>
Допустима кількість некритичних помилок (ρ) <input type="text" value="2"/>		
Пороги мають задовольняти $\theta_v \geq \theta_p \geq \theta_s$.		

Зберегти параметри
Скинути

Рис. 2.12 Сторінка моделювання перевірки документів

3 ПРОВЕДЕННЯ МОДЕЛЮВАННЯ ТА ОЦІНКА РЕЗУЛЬТАТІВ

3.1 Визначення метрик для оцінки ефективності розробленого методу.

Ефективність розробленого гібридного методу перевірки сертифікаційних авіаційних документів будемо оцінювати за сукупністю метрик, що відображають точність класифікації документа, коректність визначення змістових полів, якість виявлення аномалій графічних елементів та загальну надійність прийняття рішення про дійсність документа.

У якості оцінки результатів етапу класифікації сертифікаційних документів будемо застосовувати метрику точності класифікації (accuracy), яка відображає наскільки правильно запропонований метод визначає тип документа: «Сертифікат експлуатанта», «Схвалення організації з ТО», «Сертифікат льотної придатності», «Схвалення організації з виробництва» або «Сертифікат типу».

$$Accuracy = \frac{N_{correct}}{N_{total}} \quad (3.1)$$

де: $N_{correct}$ – кількість документів, для яких тип сертифіката було визначено правильно;

N_{total} – загальна кількість документів.

Для оцінювання якості перевірки текстових полів будемо використовувати метрику точності перевірки полів (field accuracy), яка відображає частку правильно перевірених текстових полів: номерів сертифікатів, заголовків сертифікатів, дат, тощо.

$$FieldAccuracy = \frac{F_{correct}}{F_{total}} \quad (3.2)$$

де: $F_{correct}$ - кількість текстових полів (номер сертифікату, дата, заголовок тощо) правильний формат, значення, логіку яких було вірно перевірено методом;

F_{total} – загальна кількість текстових полів, що перевірялися.

Оцінку перевірки графічних елементів будемо здійснювати за допомогою метрики косинусної подібності (cosine similarity), що оцінює подібність графічних елементів до еталонів.

$$\text{CosineSimilarity}(h, h_{ref}) = \frac{\sum_{i=1}^d h_i(h_{ref})_i}{\sqrt{\sum_{i=1}^d h_i^2} \sqrt{\sum_{i=1}^d (h_{ref})_i^2}}, \quad (3.3)$$

де: $h = (h_1, h_2 \dots, h_d)$ – embedding графічного елемента, що перевіряється;

$h_{ref} = (((h_{ref})_1, ((h_{ref})_2 \dots, ((h_{ref})_d)$ – embedding графічного елемента-зразка;

d -розмірність векторного простору.

Загальну якість підсумкової валідації сертифікаційного документу будемо оцінювати за допомогою метрики якості підсумкового рішення (document validity accuracy), яка показує, наскільки правильно запропонований метод визначає документ як Valid / Partially valid / Invalid / Suspicious/ Non-compliant / Forged.

$$\text{DocumentValidityAccuracy} = \frac{D_{correct}}{D_{total}} \quad (3.4)$$

де: $D_{correct}$ – кількість документів, фінальний статус яких визначено коректно;

D_{total} – загальна кількість документів, що перевірялися.

Запропонована сукупність метрик охоплює всі етапи роботи методу - від класифікації документу до підсумкової валідації.

3.2 Проведення моделювання та оцінка результатів

Сертифікаційні авіаційні документи є офіційними, часто закритими документами, які неможливо знайти у великій кількості у відкритому доступі. Тому, для проведення моделювання роботи пропонованого методу перевірки сертифікаційних авіаційних документів було синтезовано дані – результати аналізу документів Layout Analysis у форматі JSON.

При генерації даних використовувалися результати аналізу декількох реальних документів певного класу. Базуючись на переліку обов'язкових полів та форматів, генерувалися варіанти описів документів наступними шляхами:

- 1) додання «шуму» у координати вузлів та їх розмір;
- 2) зміни у необов'язкових вузлах: додання, видалення, зміна вмісту;
- 3) варіювання вмісту текстових полів (у рамках форматів).

Для класифікації використовувався багатошаровий перцептрон MLP з параметрами:

- 1) вхідний шар 256 нейронів (розмірність вектору ознак, отриманому після embeddings GraphSAGE);
- 2) прихований шар 128 нейронів (функція активації ReLU);
- 3) вихідний шар 5 нейронів (кількість класів документів: AOC, Part-145, CofA, Part-21G, TC);
- 4) функція активації Softmax (ймовірності, з якими документ належить до певного класу);
- 5) параметри навчання: функція втрат – Softmax Cross-Entropy, оптимізатор – Adam, Learning Rate – 0.001, Batch size – 16, Epochs – 50, розподіл train/test – 80/20.

Для моделювання процесу верифікації документів при синтезі описів документів підкреслювалась їх відношення до певного типу результату верифікації. Виконувалися маніпуляції з наступними основними параметрами об'єктів, як то структура (відсутність елементів), значення елементів (текст який містять, координати області елемента та впевненість до результатів визначення) та узгодженість (відповідність дат, ідентичність даних українською та англійською

мовою, тощо). Розглянемо зміни, що було впроваджено для опису документів-зразків з метою надання їм ознак відповідних статусів.

1. Для надання опису документа ознак впевненого статусу Valid (дійсний документ) елементам було надано високий рівень впевненості їх визначення (> 0.85), збережено наявність усіх полів та топологію їх розміщення.

2. Для надання опису документа ознак статусу Partially Valid (частково дійсний документ, або документ з зауваженнями) елементам було надано рівень впевненості визначення у діапазоні 0.4 – 0.6, додано текстових артефактів, надано зміщення області елемента, порушено формати дат.

Приклад наведено на рисунку 3.1.

```

JSON
{
  "type": "Title",
  "bbox": [148, 55, 251, 14],
  "text": "(СЕРТИФІКАТ ЕКСПЛУАТАНТА", // Помилка OCR
                                           // або помилка у документі
  "confidence": 0.55,                    // Низька впевненість
},
{
  "type": "Text",
  "bbox": [161, 611, 101, 540]
  "text": " "text": "12/05/24", // формат дати не відповідає
  "confidence": 0.63,
}

```

Рис. 3.1 Приклади надання опису документу ознак статусу Partially Valid

3. Для придання опису документа ознак статусу Invalid (недійсний документ) було видалено окремо та групами об'єкти типу «signature», «seal», «logo» (документ недійсний без печатки, підпису та/або логотипу/гербу), видалено обов'язкові для відповідного типу документу текстові елементи, змінено тексти елементів на невідповідні (за шаблоном, назва документу, тощо). Також було використано порушення логіки дат: прострочений термін та дата видачі пізніше дати закінчення. Приклад наведено на рисунку 3.2.

```

JSON
{
  "type": "Text",
  "bbox": [183,520,131,228],
  "text": " 10.01.2020", //старий документ
  "confidence": 0.66
},
{
  "type": "Title",
  "bbox": [148, 55, 251, 14],
  "text": "PART-145",
  "confidence": 0.87,
},
{
  "type": "Text",
  "bbox": [167,406,223,10],
  "text": " UA.AOC.055", //шаблон для AOC, не для PART-145
  "confidence": 0.6925
}

```

Рис. 3.2 Приклади надання опису документу ознак статусу Invalid

4. Для придання опису документа ознак статусу Suspicious (підозрілий) елементам було надано помилок, які не обов'язково відповідають ознакам підробки: невідповідні назви різними мовами або кирилиця латиниця в одному слові, невідповідний розмір печатки. Приклади наведено на рисунку 3.3.

```

JSON
{
  "type": "Title",
  "bbox": [289,371,100,9],
  "text": "АВИАЛИНИИ ХАРКОВА",
  "confidence": 0.75
},
{
  "type": "Title",
  "bbox": [193,372,82,28],
  "text": "KYIV AIRLINES", // невідповідність Харків та Київ
  "confidence": 0.6925,
}

```

Рис. 3.3 Приклади надання опису документу ознак статусу Suspicious

5. Для придання опису документа ознак статусу Forged (висока вірогідність підробки) елементам було надано конфліктуючих рівній впевненості визначення елементів. Наприклад, коли середнє значення рівній впевненості по документу (0.6-0.7) дуже відрізняється від рівня впевненості окремого елементу (0.95).

Моделювання проводилося двома експериментами.

Експеримент 1. Мета експерименту перевірка першого етапу пропонованого методу – порівняння результатів класифікації документів пропонованим підходом

та підходами TF-IDF LR та CNN. Підхід TF-IDF LR вважається класичним, щодо текстової класифікації документів. Підхід поєднує TF-IDF – перетворення тексту документу на вектор ознак та Logistic Regression – класифікація документа на основі отриманих ознак. Згорткова мережа CNN широко використовується для аналізу зображень документів. Мережа поступово формує більш складні уявлення, що дозволяє класифікувати документ за його графічним виглядом.

Експеримент проводився на заданій множині документів 20 та 40 шт зразків одного типу документу (АОС) при співвідношенні валідних документів до інших статусів як 50 до 50 та 80 до 20. Оцінка проводилася за якістю валідації (*DocumentValidityAccuracy*) та частки недійсних документів, які не було виявлено (*False Negative Rate*). У даному експерименті інші метрики, щодо перевірки тексту та графічних елементів не розглядалися, так як такі дані неможливо отримати для методів TF-IDF-LR та CNN. Отримані результати наведено у таблиці 3.1.

За результатами можна зробити висновок що пропонуванний метод дає вищу якість перевірки документів на валідність та не гіршу оцінку FNR (а при N = 40 навіть кращу).

Таблиця 3.1

Результати перевірки сертифікаційних документів методами
TF-IDF LR, CNN та GNN-методом

	Text-only TF-IDF LR	Image-only CNN	GNN метод	Text-only TF-IDF LR	Image-only CNN	GNN метод
N = 20	50/50			80/20		
Якість валідації	0.70	0.65	0.85	0.75	0.65	0.90
FNR	0.30	0.40	0.20	0.25	0.25	0,25
N = 40	50/50			80/20		
Якість валідації	0.75	0.68	0.88	0.80	0.73	0.90
FNR	0.25	0.30	0,15	0.25	0.25	0.13

Експеримент 2. Проводився на заданій множені документів 20 та 40 шт зразків різних типів документів було проведено перевірку запропонованим методом. Оцінки отримані в результаті перевірки наведено у таблиці 3.2.

Таблиця 3.2

Результати перевірки сертифікаційних документів для виборок 20 та 40

N	Точність класифікації	Якість перевірки тексту	Якість перевірки графічних	Якість валідації	FNR
20	0.90	0.85	0.84	0.90	0,14
40	0.95	0.89	0.86	0.90	0.11

За результатами перевірки можна зробити висновок, що запропонований метод дає достатньо високу якість класифікації та валідації документів. Зі збільшенням кількості документів у вибірці якість підвищується, а FNR зменшується.

Однак, загалом, можна зробити висновок, що як до сертифікаційних документів авіаційної галузі, оцінки якості та FNR виглядають недостатньо високими, що є підставою для аналізу та подальшої роботи над запропонованим методом.

ВИСНОВКИ

У результаті проведеного дослідження було проаналізовано існуючі методи, підходів та алгоритмів, що застосовуються для автоматизованої перевірки документів, визначено, що жоден окремий підхід (OCR, NLP, CV, шаблонні або криптографічні методи) не забезпечує повної перевірки сертифікаційних авіаційних документів через їхню мультимодальність та поєднання текстових, структурних, графічних і захисних елементів.

Проведено аналіз сучасного стану сертифікаційної документації у авіаційній галузі України, визначено, в Україні йде процес гармонізації авіаційного законодавства з нормами ЄС, що дозволяє безпосередньо приймати міжнародні стандартизовані форми EASA та впроваджувати регламенти EASA в українське правове поле. Досліджено специфіку сертифікаційних авіаційних документів, їх структурні особливості та визначено ключові елементи, що підлягають верифікації автентичності таких документів, а саме: стандартизовані заголовки, формати номерів сертифікатів, обов'язкові текстові поля, реєстраційні та технічні дані, а також графічні та захисні елементи, включаючи логотипи, печатки та підписи.

Розроблено гібридний метод перевірки сертифікаційних авіаційних документів, що враховує їх структурні особливості, аналіз текстових, графічних, просторових та семантичних ознак документа. Метод включає структурне моделювання макета, графову модель елементів документа, нейромережеву класифікацію та валідацію на основі rule-based logic, що забезпечує повноцінну обробку текстової та графічної інформації документа та підвищує надійність виявлення невідповідностей і ознак підробки.

Розроблено концептуальну архітектуру автоматизованої системи перевірки сертифікаційних авіаційних документів на основі клієнт-серверної моделі, здатну інтегрувати запропонований метод. Система розробляється у середовищі MS Visual Studio 2022 мовою C# з інтеграцією Python та інтерфейсом користувача на основі MVC. Розроблено програмне забезпечення, що реалізує запропонований метод.

Проведено аналіз та визначено метрики для оцінки ефективності розробленого методу: точність класифікації документу, якість перевірки текстових та графічних елементів, якість підсумкового рішення та критичну оцінку – кількість випадків невиявлення недійсного або підробленого документу. Проведено моделювання та оцінку ефективності розробленого методу перевірки сертифікаційних авіаційних документів на основі розробленого програмного забезпечення. Аналіз результатів показав, що пропонуванний метод підвищує якість перевірки на 17% в порівнянні з TF-IDF та 30% з CNN та зменшує частку документів, що помилково визначено дійсними на 30% в порівнянні з TF-IDF та 37% з CNN. Однак, показники якості перевірки 90% та частки помилково визначено дійсними документів 14%, що надає пропонуванний метод, все ще не можна назвати достатньо високими для перевірки сертифікаційних документів в авіаційній галузі, що є підставою для подальшої роботи над пропонуванним методом.

Робота пройшла апробацію на конференціях:

1. Крисевич М.А., Довженко Т.П. Класифікація авіаційних документів з використанням ШІ. VI Міжнародна науково-технічна конференція «Сучасний стан та перспективи розвитку IoT», 15 квітня 2025р., Київ, Державний університет інформаційно-комунікаційних технологій. Збірник тез. К.: ДУІКТ, 2025. С.161-163.
2. Крисевич М.А., Довженко Т.П. Визначення структури автоматизованої системи перевірки авіаційних документів. VI Всеукраїнська науково-технічна конференція «Застосування програмного забезпечення в ІКТ», 24 квітня 2025р., Київ, Державний університет інформаційно-комунікаційних технологій. Збірник тез. К.: ДУІКТ, 2025. С.304-307.

ПЕРЕЛІК ПОСИЛАНЬ

1. Cui L, Xu Y, Lv T, Wei F. Document AI: Benchmarks, Models and Applications. [Електронний ресурс]. URL: <https://arxiv.org/pdf/2111.08609>.
2. Baviskar D, Ahirrao S, Potdar V, Kotecha K. Efficient Automated Processing of the Unstructured Documents Using Artificial Intelligence: A Systematic Literature Review and Future Directions. IEEE Access. 2021. <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9402739>.
3. Ling, X, Gao M, Wang, D. Intelligent document processing based on RPA and machine learning. <https://sci-hub.se/http://dx.doi.org/10.1109/CAC51589.2020.9326579>.
4. Mandvikar S. Augmenting Intelligent Document Processing (IDP) Workflows with Contemporary Large Language Models (LLMs). International Journal of Computer Trends and Technology. 2023. <https://www.ijcttjournal.org/2023/Volume-71%20Issue-10/IJCTT-V71I10P110.pdf>.
5. Pandey M, Arora M, Arora S, AI-based Integrated Approach for the Development of Intelligent Document Management System (IDMS), Procedia Computer Science, Volume 230, 2023, pp. 725-736, <https://www.sciencedirect.com/science/article/pii/S1877050923021324>
6. Wallace P., Pum M. Intelligent Document Processing: Robotic Process Automation (RPA) and AI: Transforming Business Operations at Scale. 2024. https://www.researchgate.net/profile/Mengkorn-Pum-2/publication/390175338_Intelligent_Document_Processing_Robotic_Process_Automation_RPA_and_AI_Transforming_Business_Operations_at_Scale/links/67e35d4672f7f37c3e8d9580/Intelligent-Document-Processing-Robotic-Process-Automation-RPA-and-AI-Transforming-Business-Operations-at-Scale.pdf
7. Chai J, Zeng H, Li A, Ngai E. Deep learning in computer vision: A critical review of emerging techniques and application scenarios. Machine Learning with Applications. 2021. <http://dx.doi.org/10.1016/j.mlwa.2021.100134>.

8. Walsh J, O' Mahony N, Campbell S, Carvalho A. Deep Learning vs. Traditional Computer Vision. 2019. https://sci-hub.se/http://dx.doi.org/10.1007/978-3-030-17795-9_10.
9. Benchekroun O. Computer Vision for Document Image Analysis and Text Extraction. 2022. <https://www.diva-portal.org/smash/get/diva2:1712823/FULLTEXT01.pdf>.
10. Abirami R. N, Durai Raj Vincent P. M., Srinivasan K. Deep CNN and Deep GAN in Computational Visual Perception-Driven Image Analysis. 2021. <https://onlinelibrary.wiley.com/doi/10.1155/2021/5541134>
11. Boopathi S. Advances in Artificial Intelligence for Image Processing: Techniques, Applications, and Optimization. Handbook of Research on Thrust Technologies Effect on Image Processing. 2023. https://www.researchgate.net/publication/372921665_Advances_in_Artificial_Intelligence_for_Image_Processing_Techniques_Applications_and_Optimization
12. Жиркова, А.П. Ігнатенко, О.П. Аналіз методів машинного навчання в задачі класифікації документів. 2020. <http://dspace.nbuu.gov.ua/bitstream/handle/123456789/180496/08-Zhyrkova.pdf?sequence=1>
13. Forczmański P, Smolinski A, Nowosielski A, Małcki K. Segmentation of Scanned Documents Using Deep-Learning Approach. 2020. https://www.researchgate.net/publication/332948719_Segmentation_of_Scanned_Documents_Using_Deep-Learning_Approach.
14. Han Q, Snidauf D. Comparison of Deep Learning Technologies in Legal Document Classification. 2021. <http://dx.doi.org/10.1109/BigData52589.2021.9671486>.
15. Jiang S, Hu J, Magee C, Luo J. Deep Learning for Technical Document Classification. IEEE Transactions on Engineering Management. 2022. <http://dx.doi.org/10.1109/TEM.2022.3152216>.
16. Sachini E, Sioumalas-Christodoulou K, Christopoulos S, Karampekios N. AI for AI: Using AI methods for classifying AI science documents. Quantitative Science Studies. 2023. [Електронний ресурс]. URL: https://doi.org/10.1162/qss_a_00223.

17. Pingili R. AI-driven intelligent document processing for banking and finance. *International Journal of Management and Education in Human Development*. 2023. [Электронный ресурс]. URL: <https://www.fepbl.com/index.php/ijmer/article/view/1802>.
18. Sevim S, Omurca S.I. Ekinici E. Document Image Classification with Vision Transformers. *Lecture Notes in Computer Science*. 2022. [Электронный ресурс]. URL: https://link.springer.com/chapter/10.1007/978-3-031-01984-5_6
19. Bakkali S; Ming Z. Cross-Modal Deep Networks For Document Image Classification. *IEEE International Conference on Image Processing (ICIP)*. 2020. [Электронный ресурс]. URL: <https://scihub.se/https://ieeexplore.ieee.org/document/9191268>.
20. Audebert N., Herold C., Slimani K., Vidal C. Multimodal Deep Networks for Text and Image-Based Document Classification. *arXiv preprint*. 2019. [Электронный ресурс]. URL: <https://arxiv.org/pdf/1907.06370>
21. Jain R; Wigington C. Multimodal Document Image Classification. *International Conference on Document Analysis and Recognition (ICDAR)*. 2019. [Электронный ресурс]. URL: <https://scihub.se/https://ieeexplore.ieee.org/document/8977998>
22. Bhowmik S. Document Layout Analysis. 2023. <https://link.springer.com/book/10.1007/978-981-99-4277-0>
23. Shen Z., Zhang X., Lee J., Li J., Carlson D., Carin L. LayoutParser: A Unified Toolkit for Deep Learning Based Document Image Analysis. *arXiv preprint*. 2021. [Электронный ресурс]. URL: <https://arxiv.org/pdf/2103.15348>
24. Bhowmik S., Kundu S., Sarkar R. BINYAS: a complex document layout analysis system. *Multimedia Tools and Applications*. 2020. [Электронный ресурс]. URL: https://www.academia.edu/105046287/BINYAS_a_complex_document_layout_analysis_system
25. Pandey D., Pandey B.K., Wairya S. Hybrid deep neural network with adaptive galactic swarm optimization for text extraction from scene images. *Soft Computing*.

2020. [Электронный ресурс]. URL: <https://link.springer.com/article/10.1007/s00500-020-05245-4>

26. Sachdeva M. OCR Technology: The Cornerstone of Modern Intelligent Automation. International Journal of Information Technology and Management Information Systems. 2023. [Электронный ресурс]. URL: https://iaeme.com/MasterAdmin/Journal_uploads/IJITMIS/VOLUME_16_ISSUE_1/IJITMIS_16_01_048.pdf.

27. Sharma P. Advancements in OCR: A Deep Learning Algorithm for Enhanced Text Recognition. International Journal of Innovative Engineering and Sciences. 2023. [Электронный ресурс]. URL: <https://www.ijies.org/wp-content/uploads/papers/v10i8/F42630812623.pdf>.

28. Agarwal D, J J, Manikandan R, Ramith N, L Vandana. Advanced Automated Document Processing Using Optical Character Recognition (OCR). ResearchGate preprint. 2023. [Электронный ресурс]. URL: <http://dx.doi.org/10.1109/I2CT61223.2024.10544263>

29. Mahadevkar S, Patil S, Kotecha K, Enhancement of handwritten text recognition using AI-based hybrid approach. Journal of King Saud University - Computer and Information Sciences. 2024. [Электронный ресурс]. URL: https://www.researchgate.net/publication/389203281_Enhancement_of_handwritten_text_recognition

30. Mursari L.R, Wibowo A, The Effectiveness of Image Preprocessing on Digital Handwritten. Computer Engineering and Applications Journal. 2021. [Электронный ресурс]. URL: <http://download.garuda.kemdikbud.go.id/article.php?article=2315180&val=286&title=The%20Effectiveness%20of%20Image%20Preprocessing%20on%20Digital%20Handwritten%20Scripts%20Recognition%20with%20The%20Implementation%20of%20OCR%20Tesseract>

31. Minaee S., Kalchbrenner N. Deep Learning-based Text Classification: A Comprehensive Review. ACM Computing Surveys. 2021. [Электронный ресурс]. URL: <https://arxiv.org/pdf/2004.03705>

32. Xu Q, Gu H, Ji S. Text clustering based on pre-trained models and autoencoders. *Front Comput Neurosci*. 2024. [Электронный ресурс]. URL: <https://doi.org/10.3389/fncom.2023.1334436>

33. Herbrich R., Graepel T. Handbook of Natural Language Processing [Электронный ресурс]. URL: <https://karczmarczuk.users.greyc.fr/TEACH/TAL/Doc/Handbook%20Of%20Natural%20Language%20Processing,%20Second%20Edition%20Chapman%20&%20Hall%20Cr%20Machine%20Learning%20&%20Pattern%20Recognition%202010.pdf>.

34. Lauriola I, Lavelli A, Aioli F, An introduction to Deep Learning in Natural Language Processing: Models, techniques, and tools. *Neurocomputing*, V. 470. 2020. pp 443-456 [Электронный ресурс]. URL: <https://sci-hub.se/https://www.sciencedirect.com/science/article/abs/pii/S0925231221010997>

35. Ruiz V, Linares I, Sanchez A, Velez J.F., Off-line handwritten signature verification using compositional synthetic generation of signatures and Siamese Neural Networks. *Neurocomputing*, v 374, 2020, pp 30-41. [Электронный ресурс]. URL: <https://sci-hub.se/https://doi.org/10.1016/j.neucom.2019.09.041>.

36. Patil G, Palaiahnakote S, Gornale S, Lopresti D, Altered Handwritten Text Detection in Document Images Using Deep Learning. *International Journal of Pattern Recognition and Artificial Intelligence*. 2024. [Электронный ресурс]. URL: <http://dx.doi.org/10.1142/S0218001424520062>.

37. Castelblanco A, Solano J, Lopez C, Rivera E, Tengana Hurtado L, Ochoa M. Machine Learning Techniques for Identity Document Verification in Uncontrolled Environments: A Case Study. 2020. [Электронный ресурс]. URL: http://dx.doi.org/10.1007/978-3-030-49076-8_26.

38. Shantanu Sarode S, Khandare U, Jadhav S, Jannu A, Kamble V, Patil D, Document Manipulation Detection and Authenticity Verification Using Machine Learning and Blockchain. 2020. [Электронный ресурс]. URL: https://www.academia.edu/67784653/Document_Manipulation_Detection_and_Authenticity_Verification_Using_Machine_Learning_and_Blockchain

39. Pun A, Mohammed J.Dr., Doermann David. A Survey on Change Detection Techniques in Document Images. 2023. [Електронний ресурс]. URL: <https://arxiv.org/pdf/2307.07691>.
40. Document AI. GoogleCloud. [Електронний ресурс]. URL: <https://cloud.google.com/document-ai>.
41. Azure AI Document Intelligence. Microsoft. [Електронний ресурс]. URL: <https://azure.microsoft.com/en-us/products/ai-services/ai-document-intelligence>
42. AmazonTextract. AWS. [Електронний ресурс]. URL: https://aws.amazon.com/textract/?nc1=h_ls.
43. Autonomous transactions processing with AI agents. [Електронний ресурс]. URL: <https://www.hypatos.ai/>
44. The market-leading intelligent document processing platform. ABBYY. [Електронний ресурс]. URL: <https://www.abbyy.com/vantage/>
45. Enterprise-scale document automation, powered by AI. ABBYY. [Електронний ресурс]. URL: <https://www.abbyy.com/flexicapture>
46. Offload paperwork to AI agents and focus on. Rossum. [Електронний ресурс]. URL: <https://rosum.ai/>
47. This is somewhat embarrassing, isn't it?. Klippa. [Електронний ресурс]. URL: <https://www.klippa.com/en/doc-horizon>
48. Automated workflows for powered by AI. Nanonets. [Електронний ресурс]. URL: <https://nanonets.com/>.
49. Google Cloud Vision API. Extract insights from images, documents, and videos. Google. [Електронний ресурс]. URL: <https://cloud.google.com/vision>.
50. Azure AI Vision. Microsoft [Електронний ресурс]. URL: <https://azure.microsoft.com/en-us/products/ai-services/ai-vision>.
51. Закон України. Про ратифікацію Угоди між Україною, з однієї сторони, та Європейським Союзом і його державами-членами, з іншої сторони, про спільний авіаційний простір [Електронний ресурс]. URL: <https://zakon.rada.gov.ua/laws/show/2067-20/conv#n3>.

52. Наказ. Про затвердження Авіаційних правил України, Частина 47 «Правила реєстрації цивільних повітряних суден в Україні». [Електронний ресурс]. URL: <https://zakon.rada.gov.ua/laws/show/z1926-12#Text>.

53. Наказ № 565 Про затвердження Авіаційних правил України [Електронний ресурс]. URL: <https://zakon.rada.gov.ua/laws/show/z1056-17#Text>.

54. Державна авіаційна служба України. Сертифікати. [Електронний ресурс]. URL: <https://avia.gov.ua/npdrrd/slug-3/>.

55. Prometeia. Document Layout Analysis with Graph-based Methods. [Електронний ресурс]. URL: <https://prometeia.com/en/about-us/insights/article/document-layout-analysis-with-graphbased-methods-15929781>.

56. A Gentle Introduction to Graph Neural Networks. [Електронний ресурс]. URL: <https://prometeia.com/en/about-us/insights/article/document-layout-analysis-with-graphbased-methods-15929781>.

57. Hamri M., Devanne M. Weber J., Hassenforder M. Enhancing GNN feature modeling for document information extration using transformers. [Електронний ресурс]. URL: https://maxime-devanne.com/publis/hamri_rrpr2022.pdf.

58. Ha M.L., Blanz V. Deep Ranking with Adaptive Margin Triplet Loss. [Електронний ресурс]. URL: <https://arxiv.org/pdf/2107.06187>.

59. Крисевич М.А., Довженко Т.П. Визначення структури автоматизованої системи перевірки авіаційних документів. VI Всеукраїнська науково-технічна конференція «Застосування програмного забезпечення в ІКТ», 24 квітня 2025р., Київ, Державний університет інформаційно-комунікаційних технологій. Збірник тез. К.: ДУІКТ, 2025. С.304-307.

60. Microsoft Learn. Common web application architectures. [Електронний ресурс]. URL: <https://learn.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/common-web-application-architectures>.

61. ASP.NET MVC Guidance [Електронний ресурс]. URL: <https://learn.microsoft.com/en-us/aspnet/mvc/overview/>

ДОДАТОК А

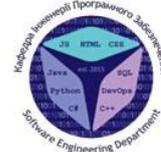
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (презентація)



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-
КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ

КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Магістерська робота

«Метод автоматизованої перевірки сертифікаційних авіаційних документів за допомогою штучного інтелекту»

Виконав: студент групи ПДМ-63 Максим КРИСЕВИЧ

Керівник: канд. техн. наук, доцент кафедри ІПЗ Тимур ДОВЖЕНКО

Київ - 2026

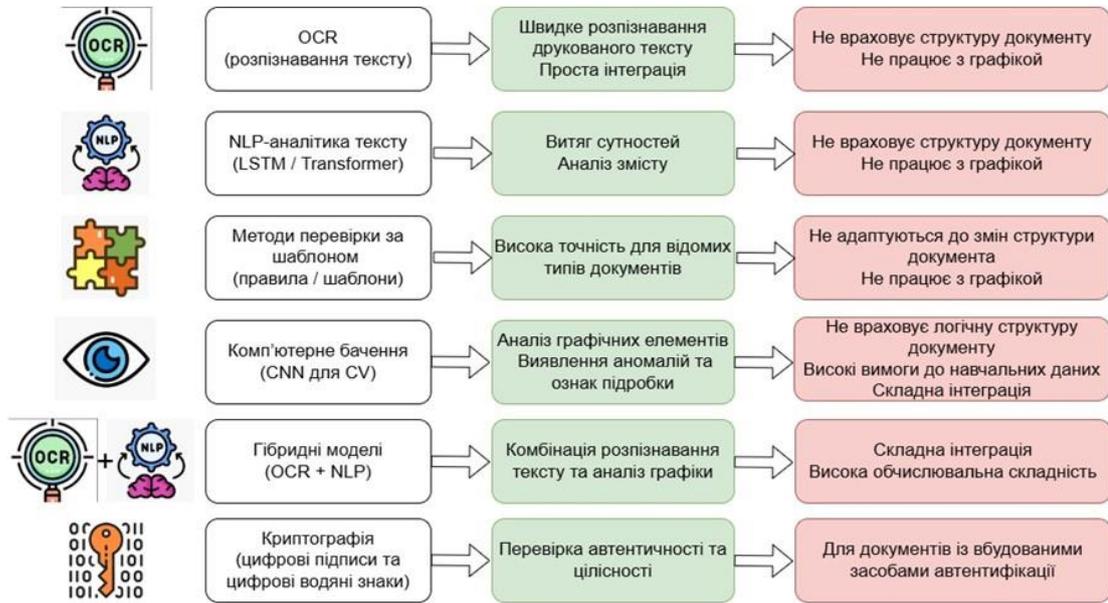
МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Мета роботи: підвищення точності перевірки та зменшення частки невиявлених недійсних сертифікаційних авіаційних документів за рахунок використання моделей та методів штучного інтелекту

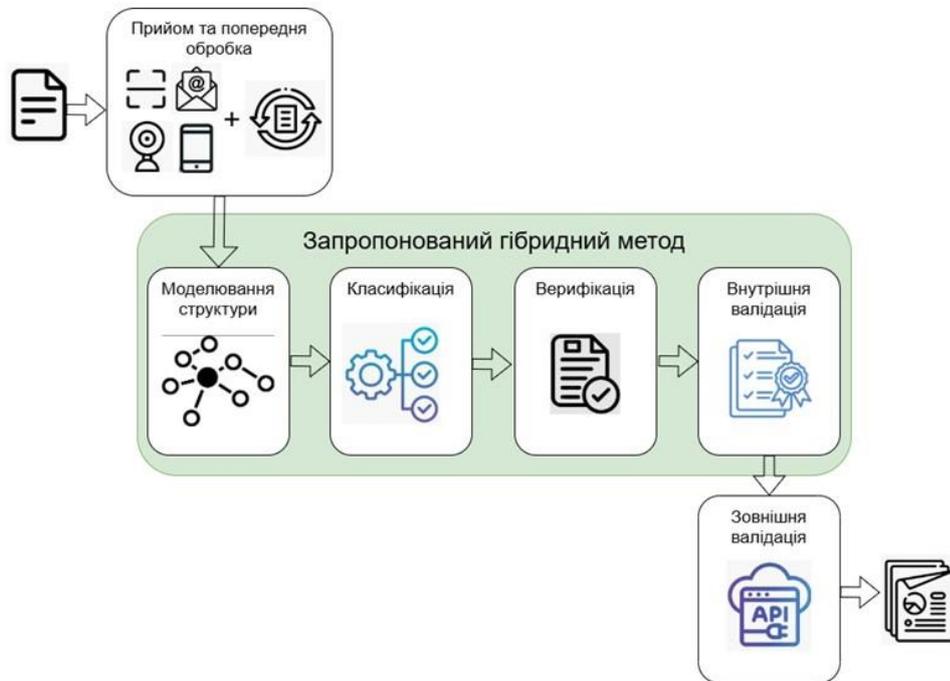
Об'єкт дослідження: процес автоматизованої перевірки сертифікаційних авіаційних документів

Предмет дослідження: моделі та методи штучного інтелекту, що застосовуються в процесі автоматизації перевірки сертифікаційних авіаційних документів.

МЕТОДИ ТА МОДЕЛІ АВТОМАТИЗОВАНОЇ ПЕРЕВІРКИ ДОКУМЕНТІВ



КОНВЕЄР АВТОМАТИЗОВАНОЇ ПЕРЕВІРКИ СЕРТИФІКАЦІЙНИХ АВІАЦІЙНИХ ДОКУМЕНТІВ



МАТЕМАТИЧНА МОДЕЛЬ МЕТОДУ ПЕРЕВІРКИ СЕРТИФІКАЦІЙНИХ ДОКУМЕНТІВ

1. Модель документа (після Layout Analysis)

$$\mathcal{E}(d) = \{e_i\}_{i=1}^n, \theta_i = (b_i, t_i, c_i, s_i, v_i)$$

де n – кількість елементів документа,
 $b_i = (x, y, w, h_i)$ – координати області елемента
 $t_i \in \mathcal{T}$ – тип елемента
 $c_i \in [0, 1]$ – довіра до детектора
 s_i – текстовий елемент
 v_i – візуальний елемент

2. Мультимодальний граф документа:

$$G_d = (V_d, E_d), \quad V_d = \{v_i | e_i \in \mathcal{E}(d)\}, |V_d| = n,$$

Структура ребер (з просторовим контекстом): $E_d = \{(i, j) | j \in kNN(i)\}$

Ознака вузла: $x_i = (p_i, c_i, o(t_i), em_i^{text}, em_i^{vis})$

де p_i – координати та розміри, c_i – довіра, $o(t_i)$ – тип елемента
 em_i^{text} – embedding текстового об'єкту
 em_i^{vis} – embedding візуального об'єкту

3. Взаування сусідів (локальний контекст):

$$h_i^{(0)} = x_i, \quad c_i^{(1)} = \text{Context}(\{h_j^{(l-1)} | j \in \mathcal{N}(i)\}),$$

$$h_i^{(l)} = \text{Update}(h_i^{(l-1)}, c_i^{(l)}), \quad h_i = h_i^{(L)}$$

де $h_i^{(0)}$ – початкова ознака вузла i , $\mathcal{N}(i)$ – множина сусідів вузла i
 h_i – кінцева ознака вузла i

4. Embedding документа: $g_d = \text{Pool}(h_1, h_2, \dots, h_n)$

5. Класифікація документа

$$z = F(g_d), \quad p = \text{softmax}(z), \quad \hat{y} = \text{argmax}_k p(c_k)$$

$$\hat{y} \in \{1, \dots, |C|\} \quad C = \{AOC, CofA, Part145, Part21G, TC\}$$

6. Семантична та логічна верифікація (rule-based):

– отримання полів документа $\hat{y} : u_d = \{(f, val_f) | f \in \mathcal{F}_{\hat{y}}\}$

де $\mathcal{F}_{\hat{y}}$ – обов'язкові поля для типу документа \hat{y}

f – назва поля за шаблоном,

val_f – реальне значення з документа.

– набір правил: $R_{\hat{y}} = \{r_1, r_2, \dots, r_k\}, r_k(u_d) \in \{0, 1\}$

– оцінка виконання правил: $S_d^{rules} = \frac{1}{K} \sum_{k=1}^K r_k(u_d)$

7. Візуальна верифікація

– візуальні елементи: $V = \{v_1, v_2, \dots, v_m\}$

– енкодер аутентичності: $z_i = f_{auth}^{(T)}(v_i)$

– attention-агрегація: $g_{vis} = \sum_{i=1}^m \alpha_i z_i, \quad \sum_{i=1}^m \alpha_i = 1$

8. Інтегральна оцінка дійсності документа

$$S_d = \lambda_1 \text{conf}_d^{cls} + \lambda_2 S_d^{rules} + \lambda_3 S_d^{graf}, \quad \lambda_i \geq 0, \quad \sum_i \lambda_i = 1$$

де: структурно-класова довіра (класифікатор): conf_d^{cls}

оцінка відповідності правилам: S_d^{rules}

графічна аутентичність: $S_d^{graf} = \frac{1}{|G|} \sum_{g \in G} I[S_g \geq \tau_g^{sus}]$

S_g – метрична оцінка елемента типу g ,

τ_g^{sus} – поріг «suspicious».

9. Фінальний статус документа

$FinalStatus(d)$

$$= \begin{cases} \text{Valid}, & S_d \geq \theta_p \cap \sum_k \delta_{d,k} = 0, \\ \text{Partially valid}, & S_d \in [\theta_p, \theta_v) \cap \sum_k \delta_{d,k} \leq \rho \\ \text{Suspicious}, & S_d \in [\theta_s, \theta_p) \cup \exists g: \text{status}_g = \text{suspicious}, \\ \text{Invalid}, & S_d < \theta_s \cup \exists k: \delta_{d,k} = 1 \text{ (critical)} \\ \text{Forged}, & \exists g: \text{status}_g = \text{forged}. \end{cases}$$

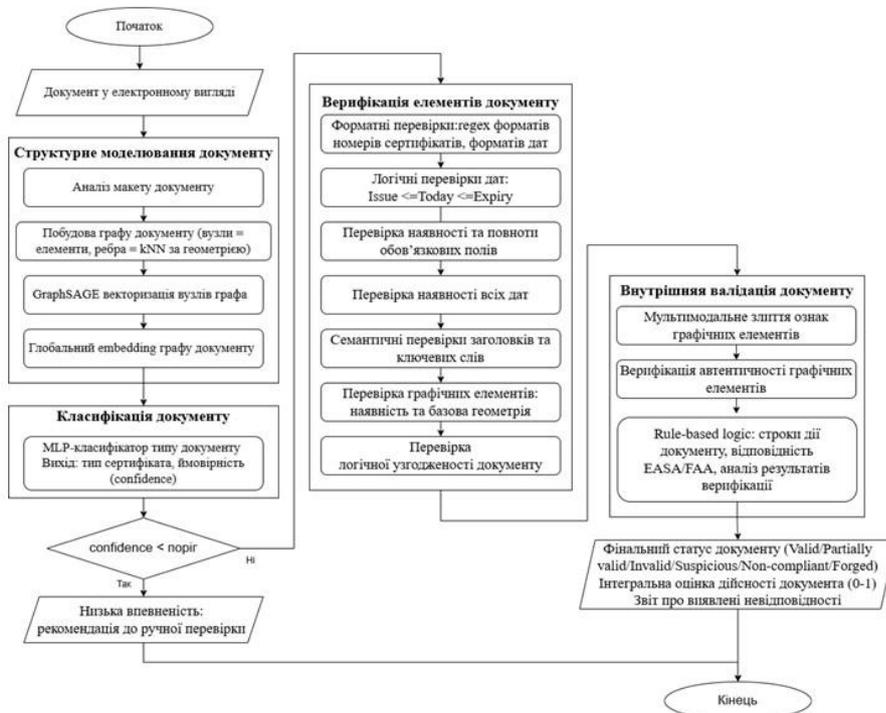
де $\delta_{d,k} \in \{0, 1\}$ – ознака критичної помилки правила k ;

$\theta_v, \theta_p, \theta_s$ – порогові значення серйозності;

ρ – допустима кількість помилок.

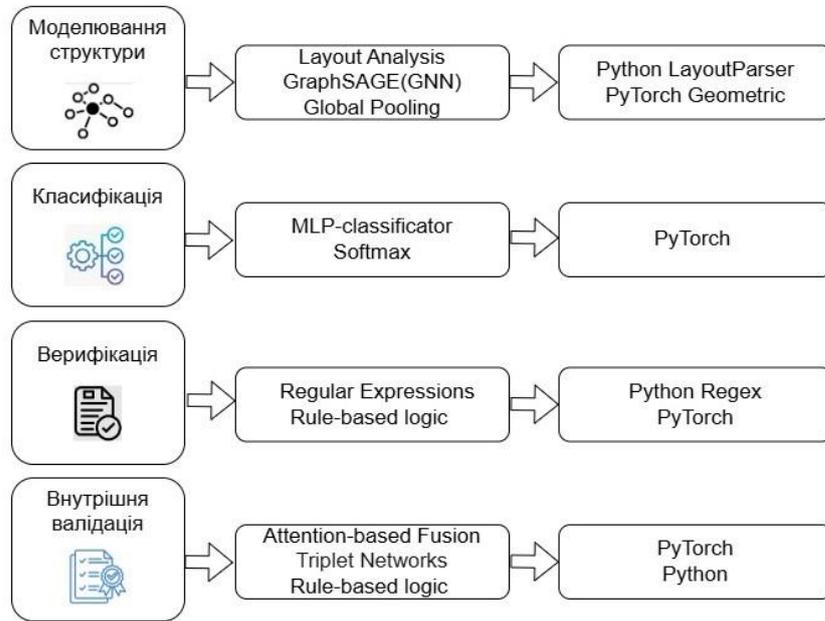
5

УЗАГАЛЬНЕНИЙ АЛГОРИТМ ГІБРИДНОГО МЕТОДУ



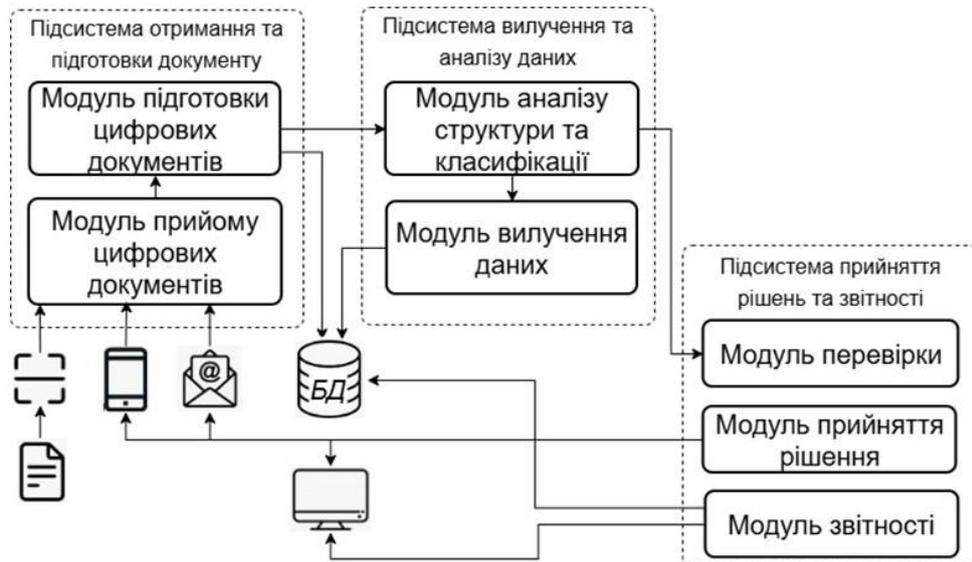
6

ТЕХНОЛОГІЇ ТА ЗАСОБИ РЕАЛІЗАЦІЇ



7

СТРУКТУРА АВТОМАТИЗОВАНОЇ СИСТЕМИ ПЕРЕВІРКИ АВІАЦІЙНИХ ДОКУМЕНТІВ



8

ПРАКТИЧНІ РЕЗУЛЬТАТИ

The screenshot displays the AviaSetVerification interface with several key sections:

- Document Status:** Shows 'PARTIALLY VALID' with an overall score of 0.78. It includes progress indicators for Classification (90%), Logic rules (96%), and Graphical elements (68%).
- Verification Parameters:** A list of parameters such as 'Формат номеру', 'Дата виходу', 'AOC дійсний', and 'Навність логотипу' with their respective values.
- Test Parameters:** Fields for document type, quantity, and verification mode.
- Quality Metrics:** A table showing 'Точність класифікації документів' (95.0%), 'Точність класифікації текстів' (89.0%), 'Точність класифікації графічних елементів' (90.0%), and 'Точність верифікації кодування' (86.0%).
- Document Results Table:**

Файл	Тип (grid)	Статус
AOC_004.jpg	AOC	Invalid
CoA_003.jpg	CoA	Valid
Part145_002.jpg	Part145	Valid
TC_004.jpg	TC	Invalid
AOC_011.jpg	AOC	Partially valid
- Verification Parameters (8-10):** A detailed table for final status:

Важк. класифікатора (k1)	Важк. правил (k2)	Важк. графічних елементів (k3)
0.4	0.35	0.25
Порі valid (θ _v)	Порі partially valid (θ _p)	Порі invalid (θ _i)
0.85	0.7	0.5

МЕТРИКИ ОЦІНЮВАННЯ ЕФЕКТИВНОСТІ ЗАПРОПОНОВАНОГО МЕТОДУ

- Точність класифікації документу : $Accuracy = \frac{N_{correct}}{N_{total}}$
де: $N_{correct}$ - кількість документів, для яких тип сертифіката було визначено правильно;
 N_{total} - загальна кількість документів.
- Якість перевірки текстових полів: $FieldAccuracy = \frac{F_{correct}}{F_{total}}$
де: $F_{correct}$ - кількість текстових полів (номер сертифікату, дата, заголовки тощо) правильний формат, значення, логіку які було вірно перевірено методом;
 F_{total} - загальна кількість текстових полів, що перевірялися.
- Якість перевірки графічних елементів: (косінусна подібність) :

$$CosineSimilarity(h, h_{ref}) = \frac{\sum_{i=1}^d h_i(h_{ref})_i}{\sqrt{\sum_{i=1}^d h_i^2} \sqrt{\sum_{i=1}^d (h_{ref})_i^2}}$$
де: $h = (h_1, h_2, \dots, h_d)$ - embedding графічного елемента, що перевіряється;
 $h_{ref} = ((h_{ref})_1, (h_{ref})_2, \dots, (h_{ref})_d)$ - embedding графічного елемента-зразка;
 d - розмірність векторного простору.
- Якість підсумкового рішення $DocumentValidityAccuracy = \frac{D_{correct}}{D_{total}}$
де: $D_{correct}$ - кількість документів, фінальний статус яких визначено коректно;
 D_{total} - загальна кількість документів, що перевірялися.
- Частка недійсних документів, які не було виявлено (помилково визначено дійсними) (False Negative Rate):

$$FNR = \frac{FN}{FN + TP}$$
де: FN (false negative) - кількість документів, які не дійсні, але визначені дійсними
 TP (true positive) - кількість документів, які не дійсні та визначені недейсними

РЕЗУЛЬТАТИ МОДЕЛЮВАННЯ

Результати перевірки сертифікаційних документів для виборок 20 та 40 при співвідношення валідних документів до невалідних 50/50 та 80/20

	Text-only TF-IDF LR	Image-only CNN	GNN метод	Text-only TF-IDF LR	Image-only CNN	GNN метод
N = 20	50/50			80/20		
Якість валідації	0.70	0.65	0.85	0.75	0.65	0.90
FNR	0.30	0.40	0.20	0.25	0.25	0.25
N = 40	50/50			80/20		
Якість валідації	0.75	0.68	0.88	0.80	0.73	0.90
FNR	0.25	0.30	0.15	0.25	0.25	0.13

Результати перевірки сертифікаційних документів GNN-методом

N	Точність класифікації	Якість перевірки тексту	Якість перевірки графічних	Якість валідації	FNR
20	0.90	0.85	0.84	0.90	0.14
40	0.95	0.89	0.86	0.90	0.11

11

ВИСНОВКИ

1. Проведено аналіз існуючих методів, підходів та алгоритмів, що застосовуються для автоматизованої перевірки документів, досліджено сертифікаційну документацію у авіаційній галузі України, визначено її специфіку.
2. Запропоновано метод перевірки документів, який ґрунтується на поданні документа у вигляді мультимодального графа («карти» документу) та реалізує на його основі комбіновану валідацію з урахуванням структурно-просторових, графічних і змістовних характеристик.
3. Спроектовано архітектуру клієнт–серверної автоматизованої системи перевірки сертифікаційних авіаційних документів та розроблено програмне забезпечення, що реалізує запропонований метод.
4. Проведено моделювання та експериментальні дослідження, що дозволили оцінити результати роботи пропонованого методу та відомих методів TF-IDF – LR та CNN методу виявлено. Аналіз результатів показав, що запропонований метод підвищує якість перевірки на 17% в порівнянні з TF-IDF – LR та 30% з CNN та зменшує частку документів, що помилково визначено дійсними на 30% в порівнянні з TF-IDF та 37% з CNN. Однак, показники якості перевірки 90% та частки помилково визначено дійсними документів 14%, що надає запропонований метод, все ще неможна назвати достатньо високими для перевірки сертифікаційних документів в авіаційній галузі, що є підставою для подальшої роботи над пропонованим методом.

12

ПУБЛІКАЦІЇ ТА АПРОБАЦІЯ РОБОТИ

Тези доповідей:

1. Крисевич М.А., Довженко Т.П. Класифікація авіаційних документів з використанням ШІ. VI Міжнародна науково-технічна конференція «Сучасний стан та перспективи розвитку IoT», 15 квітня 2025р., Київ, Державний університет інформаційно-комунікаційних технологій. Збірник тез. К.: ДУІКТ, 2025. С.161-163.
2. Крисевич М.А., Довженко Т.П. Визначення структури автоматизованої системи перевірки авіаційних документів. VI Всеукраїнська науково-технічна конференція «Застосування програмного забезпечення в ІКТ», 24 квітня 2025р., Київ, Державний університет інформаційно-комунікаційних технологій. Збірник тез. К.: ДУІКТ, 2025. С.304-307.

ДОДАТОК Б. ЛИСТИНГИ ОСНОВНИХ МОДУЛІВ

Б.1 Навчання моделей

```

# параметри
DATASET_PATH = "dataset_train"
BATCH_SIZE = 16
EPOCHS = 50
LEARNING_RATE = 0.001
TEST_SPLIT = 0.2
# класи
CLASS_MAP = {"AOC": 0, "CofA": 1,
             "Part145": 2, "Part21G": 3, "TC": 4}
def load_dataset(root_dir):
    dataset_list = []
    labels_list = []
    total_files = 0
    # прохід по класах
    for class_name, class_idx in
CLASS_MAP.items():
        folder_path = os.path.join(root_dir,
class_name)
        json_files =
glob.glob(os.path.join(folder_path, "*.json"))
        for file_path in json_files:
            try:
                with open(file_path, 'r',
encoding='utf-8') as f:
                    doc_data = json.load(f)
                    # формуємо ознаки вузлів (x)
                    img_w, img_h = 1000, 1414
                    node_features = []
                    for node in doc_data['nodes']:
                        vec =
encode node features(node, img_w, img_h,
output_dim=config.VECTOR_DIM)
                        node_features.append(vec)
                    x = torch.tensor(node_features,
dtype=torch.float)
                    # формуємо ребра (edge_index)
                    edges_source = []
                    edges_target = []
                    if 'edges' in doc_data:
                        for edge in doc_data['edges']:
edges_source.append(edge[0])
edges_target.append(edge[1])
                    if len(edges_source) > 0:
                        edge_index =
torch.tensor([edges_source, edges_target],
dtype=torch.long)
                    else:
                        edge_index =
torch.tensor([[0], [0]], dtype=torch.long)
                    # мітки
                    y = torch.tensor([class_idx],
dtype=torch.long)
                    data = Data(x=x,
edge_index=edge_index, y=y)
                    dataset_list.append(data)
                    labels_list.append(class_idx)
                    total_files += 1
            return dataset_list, labels_list

def train_gnn_model():
    # завантажуюмо дані
    full_dataset, labels =
load_dataset(DATASET_PATH)

    if len(full_dataset) < 5:
        return

# розподіляємо тренув/тестув (80/20)
train_data, test_data = train_test_split(
    full_dataset,
    test_size=TEST_SPLIT,
    stratify=labels,
    random_state=42
)
train_loader = DataLoader(train_data,
batch_size=BATCH_SIZE, shuffle=True)
test_loader = DataLoader(test_data,
batch_size=BATCH_SIZE, shuffle=False)
# ініціалізуємо модель
model =
DocumentGNN(input_dim=config.VECTOR_DIM,
num_classes=len(CLASS_MAP))
# оптимізатор та лосс
optimizer = optim.Adam(model.parameters(),
lr=LEARNING_RATE)
# CrossEntropyLoss він вже з LogSoftmax +
NLLLoss
criterion = nn.CrossEntropyLoss()
# навчаємо
for epoch in range(EPOCHS):
    model.train()
    total_loss = 0
    correct_train = 0
    total_train = 0
    for batch in train_loader:
        optimizer.zero_grad()
        # forward pass
        logits, probs = model(batch.x,
batch.edge_index, batch.batch)
        loss = criterion(logits, batch.y)
        loss.backward()
        optimizer.step()
        total_loss += loss.item()
        # точність
        pred = logits.argmax(dim=1)
        correct_train += int((pred ==
batch.y).sum())
        total_train += batch.y.size(0)
    avg_loss = total_loss / len(train_loader)
    train_acc = correct_train / total_train
    # перевірка на тест 20
    model.eval()
    correct_test = 0
    total_test = 0
    with torch.no_grad():
        for batch in test_loader:
            logits, probs = model(batch.x,
batch.edge_index, batch.batch)
            pred = logits.argmax(dim=1)
            correct_test += int((pred ==
batch.y).sum())
            total_test += batch.y.size(0)
    test_acc = correct_test / total_test if
total_test > 0 else 0
    # зберігаємо навчану модель
    os.makedirs(config.MODELS_DIR, exist_ok=True)
    save_path = os.path.join(config.MODELS_DIR,
"gnn_model.pth")
    torch.save(model.state_dict(), save_path)

class TripletMockDataset(Dataset):
    def __init__(self, size=100):
        self.size = size
    def __len__(self):

```

```

        return self.size
    def getitem (self, idx):
        anchor = torch.randn(3, 64, 64) # (3
каналы, 64x64 пікселі)
        # Positive схожий але трохи шуму
        positive = anchor + (torch.randn(3, 64,
64) * 0.1)
        # Negative зовсім інший
        negative = torch.randn(3, 64, 64)
        return anchor, positive, negative
# навчання
def train visual_model():
    EPOCHS = 10
    BATCH SIZE = 8
    MARGIN = 1.0 # зазор між позитивним та
негативним прикладом
    dataset = TripletMockDataset(size=100)
    loader = DataLoader(dataset,
batch size=BATCH SIZE, shuffle=True)
    model = VisualAuthenticator()
    optimizer = optim.Adam(model.parameters(),
lr=0.001)

```

```

        criterion =
nn.TripletMarginLoss(margin=MARGIN, p=2)
        model.train()
        for epoch in range(EPOCHS):
            running_loss = 0.0
            for anchor, positive, negative in loader:
                optimizer.zero_grad()
                # отримуємо вектори зображень
                emb_a = model(anchor)
                emb_p = model(positive)
                emb_n = model(negative)
                loss = criterion(emb_a, emb_p, emb_n)
# пропрацуємо loss
                loss.backward()
                optimizer.step()
                running_loss += loss.item()
            # зберігаємо навчану модель
            os.makedirs(config.MODELS DIR, exist ok=True)
            save_path = os.path.join(config.MODELS_DIR,
"visual_net.pth")
            torch.save(model.state_dict(), save_path)

```

Б.2 Визначення елементів структури

```

def run layout analysis(image path):
    if not os.path.exists(image_path): # файл не
знайдено
        return []
    try:
        image =
cv2.imdecode(np.fromfile(image path,
dtype=np.uint8), cv2.IMREAD_COLOR)
        if image is None: # не вдалося прочитати
зображення
            return []
        image_rgb = cv2.cvtColor(image,
cv2.COLOR_BGR2RGB)
        except Exception as e: # помилка відкриття
            return []
        layout_data = []
        layout_result = model.detect(image_rgb) #
детекція блоків
        for block in layout_result:
            if block.score < 0.5:
                continue
            x1, y1, x2, y2 = map(int,
block.coordinates)
            w = x2 - x1
            h = y2 - y1
            h_img, w_img, _ = image.shape # перевірка
координат
            x1, y1 = max(0, x1), max(0, y1)
            x2, y2 = min(w_img, x2), min(h_img, y2)
            roi = image_rgb[y1:y2, x1:x2]
            text_content = ""
            if block.type in ["Text", "Title",
>List"]:
                try:
                    text_content =
pytesseract.image_to_string(roi, lang='ukr+eng',
config='--psm 6').strip()
                except Exception as e: # ocr error
                    return []
            layout_data.append({
                "type": block.type,
                "bbox": [x1, y1, w, h],
                "text": text_content,
                "confidence": float(block.score),
                "field": "unknown"
            })
# шукаємо графіку
visual_elements = detect_blue_regions(image)

```

```

        layout_data.extend(visual elements)
        layout_data.sort(key=lambda x: x['bbox'][1])
        save_json(layout_data, "1_layout.json")
        return layout_data

def detect_blue_regions(image):
    found elements = []
    hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
    lower_blue = np.array([90, 50, 50])
    upper_blue = np.array([140, 255, 255])
    mask = cv2.inRange(hsv, lower_blue,
upper blue)
    kernel =
cv2.getStructuringElement(cv2.MORPH_RECT, (10,
10))
    mask_dilated = cv2.dilate(mask, kernel,
iterations=2)
    contours, _ = cv2.findContours(mask_dilated,
cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
    for cnt in contours:
        x, y, w, h = cv2.boundingRect(cnt)
        area = w * h
        if area < 500 or area > (image.shape[0] *
image.shape[1] * 0.5):
            continue
        aspect_ratio = w / float(h)
        if 0.8 <= aspect_ratio <= 1.3:
            e_type = "Stamp"
        elif aspect_ratio > 1.3:
            e_type = "Signature"
        else:
            e_type = "Logo"
        found_elements.append({
            "type": e_type, "bbox": [x, y, w, h],
            "text": "",
            "confidence": 0.85, "field":
"visual_element"
        })
    return found_elements

if os.path.exists(test_file):
    run_layout_analysis(test_file)

```

Б.3 Класифікація

```

def run():
    # завантажуюємо дані
    layout_data =
load json(f"{config.ARTIFACTS_DIR}/1_layout.json")
    # розмір з layout metadata)
    IMG_W, IMG_H = 1000, 1414
    nodes = []
    centers = []
    node_features = []
    # вузли та ознаки
    for item in layout_data:
        # векторизуємо
        feat_vec = encode_node_features(item,
IMG W, IMG H, output dim=config.VECTOR_DIM)
        node_features.append(feat_vec)
        # центр елемента (для kNN та відстані)
        bbox = item['bbox']
        centers.append([bbox[0] + bbox[2]/2,
bbox[1] + bbox[3]/2])

        # зберігаємо вузол з вектором
        item['vector'] = feat_vec[:5] + "..."
        nodes.append(item)

    # ребра
    edges = []
    edge_index_tensor = [[], []]
    k = config.GRAPH_K_NEIGHBORS
    if len(centers) > k:
        nbrs =
NearestNeighbors(n_neighbors=k+1).fit(centers)
        distances, indices =
nbrs.kneighbors(centers)
        for i, neighbors in
enumerate(indices):
            for n_idx in neighbors[1:]:
                edges.append([i, int(n_idx)])
                edge_index_tensor[0].append(i)
                edge_index_tensor[1].append(n_idx)

    # запускаємо GNN класифікація

    doc_type = "Unknown"
    confidence = 0.0
    if config.MOCK_MODELS:
        # doc type = "AOC"
        # confidence = 0.92
    else:
        x = torch.tensor(node_features,
dtype=torch.float)
        edge_index =
torch.tensor(edge_index_tensor, dtype=torch.long)
        batch = torch.zeros(x.size(0),
dtype=torch.long)
        model = DocumentGNN(config.VECTOR_DIM, 64,
5)
        model.load_state_dict(torch.load("models/weights/g
nn.pth"))
        model.eval()
        with torch.no_grad():
            gd, probs = model(x, edge_index,
batch)

        classes = ["AOC", "CofA", "Part145",
"Part21G", "TC"]
        pred_idx = torch.argmax(probs).item()
        doc_type = classes[pred_idx]
        confidence = probs[0][pred_idx].item()

    # зберігаємо результат
    graph_output = {
        "nodes": nodes,
        "edges": edges,
        "doc type": doc_type,
        "classifier_conf": confidence
    }
    save_json(graph_output,
f"{config.ARTIFACTS_DIR}/2_graph.json")

    run()

```

Б.4 Верифікація

```

transform_pipeline = T.Compose([
    T.ToPILImage(),
    T.Resize((64, 64)),
    T.ToTensor(),
    T.Normalize(mean=[0.485, 0.456, 0.406],
std=[0.229, 0.224, 0.225])
])

def get_visual_embedding(model, image_crop):
    # векторні ознаки
    if image_crop is None or image_crop.size == 0:
        return None
    tensor = transform_pipeline(image_crop)
    tensor = tensor.unsqueeze(0)
    with torch.no_grad():
        embedding = model.forward_one(tensor)
    return embedding.numpy().flatten()

def load_visual_model():
    # Triplet
    model = VisualAuthenticator()
    weight_path = os.path.join(config.MODELS_DIR,
"visual_net.pth")
    if os.path.exists(weight_path):
        model.load_state_dict(torch.load(weight_path))

        model.eval()
        return model

CERT PATTERNS = {
    "AOC": r"[A-Z]{2}\.AOC\.\d{3}", #
    "UA.AOC.001", #
    "Part145": r"[A-Z]+\.\d{4}", #
    "EASA.145.1234", #
    "Part21G": r"[A-Z]+\.\d{4}", #
    "EASA.21G.1234", #
    "TC": r"[A-Z]+\.[A-Z]\.\d{3}", #
    "EASA.A.064", #
    "CofA": r"UA\.\d{4}" #
}

def extract_dates(text):
    # пошук дат DD.MM.YYYY та YYYY-MM-DD
    date_pattern =
r"\b(\d{2})[./](\d{2})[./](\d{4})\b"
    matches = re.findall(date_pattern, text)
    dates = []
    for day, month, year in matches:
        try:
            d = datetime(int(year), int(month),
int(day))

```

```

        dates.append(d)
    except ValueError:
        continue
    return sorted(dates)

def check_logical_rules(doc_data):
    # перевірка правил
    doc_type = doc_data.get('doc_type', 'Unknown')
    nodes = doc_data['nodes']
    full_text = " ".join([n.get('text', '') for n
in nodes])
    issues = []
    rules_total = 0
    rules_passed = 0

    current_date = datetime.now()
    # обов'язкові
    rules_total += 1
    # сертифікат
    pattern = CERT_PATTERNS.get(doc_type)
    if pattern and re.search(pattern, full_text):
        rules_passed += 1
    elif not pattern:
        if re.search(r"\d{3,}", full_text):
            rules_passed += 1
        else:
            issues.append("Missing or invalid
Certificate Number")
        else:
            issues.append(f"Certificate Number does
not match pattern for {doc_type}")
        # дата видачі
        dates = extract_dates(full_text)
        rules_total += 1
        if dates:
            rules_passed += 1
            issue_date = dates[0] #
        else:
            issues.append("Missing Issue Date (No
dates found)")
            issue_date = None
        # логічна часова
        if issue_date:
            # дата видачі <= поточна
            rules_total += 1
            if issue_date <= current_date:
                rules_passed += 1
            else:
                issues.append(f"Issue Date is in the
future: {issue_date.strftime('%d.%m.%Y')}")
            # спеціфічні
            if doc_type == "АОС":
                # чинним до внесення змін (безстроковий)
                pass
            elif doc_type == "Part145":
                # якщо Part145 без Rating категорій ->
                Non-compliant
                rules_total += 1
                if "Rating" in full_text or "Class" in
full_text or "Maintenance" in full_text:
                    rules_passed += 1
                else:
                    issues.append("Part-145: Missing
'Rating' or 'Maintenance' categories")
            elif doc_type == "Part21G":
                # Part-21G є чинним, допоки не буде
                скасовано
                rules_total += 1
                if "unlimited" in full_text.lower() or
"cancelled" in full_text.lower():
                    rules_passed += 1
                else:
                    issues.append("Part-21G: Missing
validity statement (Unlimited/Until cancelled)")

            elif doc_type == "CofA":
                # якщо CofA без registration marks ->
                недійсний
                rules_total += 1

                if "Registration" in full_text or "Mark"
in full_text or re.search(r"UR-[A-Z]{3,}",
full_text):
                    rules_passed += 1
                else:
                    issues.append("CofA: Missing Aircraft
Registration Marks")
                    # CofA чинний впродовж року
                    if issue_date:
                        rules_total += 1
                        expiry_date = issue_date +
timedelta(days=365)
                        if current_date <= expiry_date:
                            rules_passed += 1
                    else:
                        issues.append("CofA: Document
Expired (>12 months from Issue Date)")

                    elif doc_type == "TC":
                        # модель - серійний номер
                        rules_total += 1
                        if "Model" in full_text or "Type" in
full_text:
                            rules_passed += 1
                        else:
                            issues.append("TC: Missing Aircraft
Model/Type information")

                        # наявність підпису, печатки
                        visual_nodes = [n for n in nodes if n['type']
in ['Stamp', 'Signature']]
                        rules_total += 1
                        if len(visual_nodes) > 0:
                            rules_passed += 1
                        else:
                            issues.append("Missing structural element:
Stamp or Signature")
                            # пахуємо Sd rules
                            sd_rules = rules_passed / rules_total if
rules_total > 0 else 0.0
                            return sd_rules, issues

def check_visual_auth(doc_data,
original_image_path):
    # графічні Sdgraf
    visuals = [n for n in doc_data['nodes'] if
n['type'] in ['Stamp', 'Signature', 'Logo']]
    if not visuals:
        return 0.5, ["No visual elements found"]
    # завантажуюмо зображення
    if not os.path.exists(original_image_path):
        return 0.0, ["Image file not found:
{original_image_path}"]
    stream = open(original_image_path, "rb")
    bytes = bytearray(stream.read())
    numpyarray = np.asarray(bytes, dtype=np.uint8)
    full_image = cv2.imdecode(numpyarray,
cv2.IMREAD_COLOR)
    # завантажуюмо модель
    model = load_visual_model()
    valid_cnt = 0
    issues = []
    reference_embedding = np.random.rand(128)
    reference_embedding = reference_embedding /
np.linalg.norm(reference_embedding)
    img_h, img_w, _ = full_image.shape
    for vis in visuals:
        bbox = vis.get('bbox')
        if not bbox: continue
        x, y, w, h = map(int, bbox)
        x, y = max(0, x), max(0, y)
        w, h = min(img_w - x, w), min(img_h - y,
h)

        # вирізання
        crop = full_image[y:y+h, x:x+w]
        embedding = get_visual_embedding(model,
crop)
        if embedding is None:

```

```

        issues.append(f"Failed to process
visual element: {vis['type']}")
        continue
        # оцінка схожості
        similarity = 1 - cosine(embedding,
reference_embedding)

        THRESHOLD = 0.6
        if similarity >= THRESHOLD:
            valid_cnt += 1 # [cite: 62]
        else:
            issues.append(f"Suspicious
{vis['type']} (similarity: {similarity:.2f})")
            # Sdgraf
            sd_graf = valid_cnt / len(visualls) if visualls
        else 0.0
        return sd_graf, issues

def run():
    graph_path =
os.path.join(config.ARTIFACTS_DIR, "2 graph.json")
    image_path = os.path.join("input_docs",
"sample cert.jpg")
    data = load_json(graph_path)
    # вхідні оцінки
    conf_cls = data.get('classifier_conf', 0.5)
    # логічна перевірка
    sd_rules, rule_issues =
check_logical_rules(data)
    # візуальна перевірка
    sd_graf, graf_issues = check_visual_auth(data,
image_path)

    # інтегральна оцінка Sd
    sd_final = (config.LAMBDA_1 * conf_cls) + \
                (config.LAMBDA_2 * sd_rules) + \
                (config.LAMBDA_3 * sd_graf)
    # кінцевий статус
    critical_errors = [i for i in rule_issues if
"Invalid" in i or "Missing" in i]

    if critical_errors:
        status = "Invalid"
    elif sd_final >= config.THRESHOLD_VALID:
        status = "Valid"
    elif sd_final >= config.THRESHOLD_SUSPICIOUS:
        status = "Partially Valid"
    else:
        status = "Suspicious"
    print(f"SCORES: Cls={conf_cls:.2f},
Rules={sd_rules:.2f}, Graf={sd_graf:.2f}")
    print(f"FINAL SCORE (Sd): {sd_final:.4f} ->
(status)")
    report = {
        "doc_type": data.get('doc_type',
'Unknown'),
        "final_status": status,
        "integral_score": sd_final,
        "issues": rule_issues + graf_issues
    }
    save_json(report,
os.path.join(config.ARTIFACTS_DIR,
"3_report.json"))

```