AMOUNT INFLATION FRAUD

Goal: Identify any mismatch between actual line item totals and the declared total amount on the invoice.

ALL POSSIBLE WAYS FRAUD CAN OCCUR (Amount Inflation):

Manual editing of the Total field	Changing the final total on the invoice image after generation.
Tampering with unit prices or quantities	Inflating line items to raise total but keeping product/service description same
Adding invisible/blank line items	Hidden or blank entries to justify a higher total (often seen in PDFs)
Swapping currency labels	E.g., changing 5,000 THB to 5,000 PLN (which is 7× more)
Falsified discount/fee entries	Adding fake fees to boost totals or removing actual discounts

HOW TO DETECT THESE FRAUD PATTERNS

Detection Strategy	
Sum Validation	Calculate the sum of (unit price × quantity) for all line items and compare it to the stated total.
Threshold Check	Allow a small margin (e.g., ±5 PLN). Any difference beyond that is flagged.
Field Consistency Rules	Ensure currency is consistent across line items and total.
Fee/Discount Logic	If fees are present, ensure they are labeled, positive, and match company policy.
Visual Region Analysis (Advanced)	If available, scan the total field area on the invoice image for tampering (e.g., fonts/pixel issues).

HOW WE EXTRACT THE NECESSARY FIELDS

To detect amount inflation, we have to extract:

Field	Extraction Method
`line_items`	Use OCR and predefined field templates (e.g., extract all rows under “Description”, “Qty”, “Unit Price”)
`total_amount`	Locate the “Total” or “Amount Due” field using OCR + keyword-matching
`currency`	Extract currency symbol near the total or unit price (PLN, EUR, $, etc.)
`tax/fees`	Capture lines labeled as "Tax", "VAT", "Service Fee", "Discount" if present

DATE MANIPULATION FRAUD

Goal: Detect if the **invoice or service date** has been **intentionally altered** to make a claim valid when it shouldn't be like outside contract period, tax reporting period, or budget window

ALL POSSIBLE WAYS DATE MANIPULATION CAN OCCUR

Fraud Tactic	Description
Future-dated invoice	Invoice date is set in the future to delay payment or confuse systems.
Backdated invoice	Invoice date is moved backward to fit into a closed reporting or tax period.
Service date misalignment	Service was delivered in March, but invoice date is in February.
Falsified contract timeline	Invoice reflects a date outside the agreed contract or delivery period.
Fake due date manipulation	Payment due date is shortened or extended unnaturally.

HOW TO DETECT THESE FRAUD PATTERNS

Detection Strategy	How It Works
Future Date Rule	Flag if the invoice date is after today's date.
Contract Window Validation	Compare invoice or service date with allowed contract timeline.
Cross-field consistency check	Ensure `service_date` ≤ `invoice_date` ≤ `due_date`.
Multiple invoice patterns	Detect irregular billing frequency (e.g., suddenly 2 invoices in same week).
Business calendar alignment	Check if invoice was issued on an illogical day (e.g., holiday or weekend).

HOW WE EXTRACT THE NECESSARY FIELDS

For this fraud, we extract:

Field	Extraction Method
`invoice_date`	Look for terms like “Invoice Date”, “Issued On” using **OCR**
`service_date`	Terms like “Date of Service”, “Delivery Date” (if present)
`due_date`	Terms like “Due Date”, “Pay By”
`today's date`	System clock (dynamic)
`contract period`	From internal system or predefined project data

WHAT THE INVOICE FORMAT LOOKS LIKE

A typical date section in an invoice:

Invoice No: INV-2025-0532
Invoice Date: 2025-06-25
Service Period: 2025-04-01 to 2025-04-30
Due Date: 2025-07-10

In a fraud version:

- Invoice Date might be set to **2025-04-25** instead of 2025-06-25 to fit a March budget.
- Or Due Date is set too early (e.g., next day), forcing urgent processing.

COMMON DATE FORMATS WE MUST HANDLE

Format	Example	Used In
`YYYY-MM-DD`	2025-04-28	ISO standard, databases
`DD-MM-YYYY`	28-04-2025	Common in Europe, incl. Poland
`MM/DD/YYYY`	04/28/2025	United States
`DD Month YYYY`	28 April 2025	Formal printed invoices
`DD.MM.YYYY`	28.04.2025	Central Europe, Eastern Europe
`DD/MM/YY`	28/04/25	Short form used in receipts
`Month DD, YYYY`	April 28, 2025	Common in business reports (US)

How We Handle These:

- Use intelligent **date parsers** that:
  - Detect different delimiters (`-`, `/`, `.`, space)
  - Interpret day/month ordering based on locale context
- Normalize all dates into a **single format internally**:
  Example: `YYYY-MM-DD` (e.g., `2025-04-28`)

If we don’t normalize:

- A date like `04/05/2025` could mean:
  - **May 4, 2025** (US)
  - **April 5, 2025** (Europe)
    → A fraudster might use this ambiguity to deceive.

Can applying accurate rules like:

- Is the invoice issued **before** or **after** service?
- Is the invoice **dated in the future**?
- Is the **due date too early or too late**?

DUPLICATE SUBMISSION FRAUD

Goal: Detect if the same invoice is submitted more than once possibly with small edits like a changed invoice number or date to get paid twice.

ALL POSSIBLE WAYS DUPLICATE SUBMISSION CAN OCCUR

Fraud Tactic	Description
Exact duplicate	Same invoice submitted twice without any changes.
Invoice number changed	Invoice is cloned, but number is slightly altered (e.g., INV-0421 → INV-0421A).
Date changed	Same invoice, different date to look like a new one.
Amount slightly changed	Increase by 1–2 PLN to bypass exact match checks.
Vendor name variation	Uses a slightly different spelling or formatting (e.g., "ABC Ltd." vs "A.B.C. Ltd").

HOW TO DETECT THESE FRAUD PATTERNS

Detection Strategy	How It Works
Invoice fingerprinting	Combine key fields into a unique “fingerprint” (e.g., supplier + date + amount + items).
Fuzzy matching	Check if a new invoice is *highly similar* to a previous one, even if some fields differ.
Historical lookup	Maintain a database of processed invoice fingerprints to compare new entries against.
Text similarity score	Use NLP techniques (or basic string matching) to catch small variations.

HOW WE EXTRACT THE NECESSARY FIELDS

Field	How It's Used
`invoice_number`	Compare directly or check for subtle modifications (e.g., “A” added).
`supplier_name`	Standardize to remove spacing/punctuation inconsistencies.
`invoice_date`	Used to check if the document is spaced realistically from the previous one.
`line_items`	Extract and hash the structure or contents.
`total_amount`	Directly compared or included in invoice fingerprint.

→ After extracting, we generate a **fingerprint** like:

```
"ABC Ltd + 2025-04-20 + 5000.00 + Laptop-Monitor"
```

Then compare that to previously stored fingerprints to spot duplicates.

EXAMPLE
WHAT A DUPLICATE INVOICE MIGHT LOOK LIKE

Original:

Invoice No: INV-2025-0421
Supplier: ABC Ltd
Date: 2025-04-20
Total: 5,000 PLN

Duplicate (Fraud):

Invoice No: INV-2025-0421A
Supplier: A.B.C. Ltd
Date: 2025-05-15
Total: 5,000 PLN

MISSING OR INVALID FIELDS (Anomaly Detection)

Goal: Detect if any **mandatory fields** are missing or incorrectly formatted in the invoice. This can indicate carelessness, fraud attempts, or OCR failure.

ALL POSSIBLE WAYS THIS CAN HAPPEN

Anomaly Type	Description
Missing invoice number	No unique ID makes it difficult to trace or detect duplicates
Invalid or missing Tax ID/VAT	Fake or blank taxpayer identification number – common in supplier fraud
Missing supplier name	Supplier identity is unclear or deliberately hidden
Invalid bank account number	Bank account doesn't match standard format (e.g., wrong IBAN)
OCR extraction failure	Field exists in image but was not captured due to font, resolution, or layout

HOW TO DETECT THESE PROBLEMS

Detection Strategy	How It Works
Field presence validation	Check that required fields exist and are not blank
Format validation (Regex)	Check field content against known patterns (e.g., Polish NIP, IBAN)
Field consistency check	Ensure invoice number and tax ID follow supplier's usual format
Database match (optional)	Validate tax ID or bank account against registered supplier records

FIELDS TO EXTRACT + FORMATS TO CHECK

Field	Expected Format	Validation Rule
`invoice_number`	Alphanumeric, min 3–5 characters	Must exist and be unique; no repeats
`tax_id` / `VAT`	Poland: 10-digit number (NIP)	Must match regex: `^\d{10}$`
`supplier_name`	String, 2+ characters	Must not be empty
`bank_account`	IBAN (e.g., PLkk BBBB BBBB CCCC CCCC CCCC CC)	Must match regex: `^PL\d{26}$` (for Poland)
`total_amount`	Numeric with 2 decimal places	Must be positive and formatted as currency

EXAMPLE PROBLEMATIC INVOICE MIGHT LOOK LIKE

Invoice No:
Supplier: ABC Company
Tax ID: 12O456789O ← "O" instead of "0"
Bank Account: PL12 3456 7890 1234 ← Too short

3 Mains flags would trigger:

- Missing invoice number
- Invalid tax ID (bad characters)
- Invalid bank account (wrong format)

VISUAL TAMPERING DETECTION (Fraud)

Goal: Detect if parts of the document image have been **visually edited or manipulated**, such as changing the amount, adding a fake signature, or modifying supplier details.

ALL POSSIBLE WAYS VISUAL TAMPERING CAN OCCUR

Tampering Type	Description
Edited amount	Total amount or unit price changed using a photo editor
Fake or copied signature	Signature added from a different document or drawn in
Deleted information	Fields erased or blurred out, like invoice number or tax ID
Stamp/sticker overlays	A stamp image is pasted over a section to cover it or falsely approve it
Fonts inconsistent	Part of the text uses different font, alignment, or sharpness

HOW TO DETECT THESE FRAUD PATTERNS

Detection Strategy	How It Works
Error Level Analysis (ELA)	Measures compression inconsistencies; tampered areas show different error levels
Visual signature verification	Compares extracted signature with previous known signatures (if available)
Image region analysis (CNN)	A deep learning model classifies regions (like total or signature) as real or fake
Font/texture analysis	Checks for mismatched font style, spacing, or pixel clarity in key fields

WHAT WE EXTRACT & ANALYZE

To detect tampering, we focus on image regions:

Region	Purpose
`total_amount`	Check for font mismatch, copy-paste edits
`signature area`	Detect visual inconsistency or fake signature
`invoice number`	Often edited or erased in fraudulent copies
`stamp area`	Detect overlay images or stickers

EXAMPLE TAMPERED INVOICE MIGHT LOOK LIKE

Edited Image:

- The “Total Amount” field looks sharp and pixelated while the rest of the document is smooth.
- Signature has no pen pressure variation or inconsistent alignment.
- Font in the amount is bolder and not aligned with other numbers.