**Tamper Detection in Academic Credentials**

**Internship Role:** AI Development Intern

**Candidate:** Umesh Chandra Bojja

**Project Title:** Tamper Detection in Academic Credentials

**Submission:** GitHub + Google Drive

**Objective:**

The goal of this project is to develop a lightweight, automated system to detect tampering in academic credentials, with a focus on Great Learning certificates. This includes detecting inconsistencies in user-submitted certificate details and flagging invalid or manipulated entries.

**Approach, Assumptions, and Methodology**

**Assumptions:**

* Valid certificates follow a consistent structure and value range:
  + Certificate IDs follow the format GL-XXXXX
  + Course names belong to a known list (e.g., AI Foundations, Machine Learning, etc.)
  + Issue dates fall within a reasonable academic range (2020–2025)
* Names are considered valid if they have appeared in past authentic entries.

**Methodology:**

* **Data Simulation**: A dataset was generated with both valid and tampered certificate entries.
* **Feature Engineering**:
  + valid\_id\_format: 1 if ID format matches GL-XXXXX, else 0
  + valid\_course: 1 if course exists in approved list
  + valid\_date: 1 if year is between 2020 and 2025
  + name\_encoded: Encoded value of the user’s name (for consistency)
* **Model Selection**:
  + A **Random Forest Classifier** was trained using scikit-learn on this feature set.
* **Prediction Logic**:
  + User enters certificate info; features are computed dynamically; the model returns Valid or Tampered.

**Challenges and Trade-offs:**

|  |  |
| --- | --- |
| **Challenge** | **Description** |
| **Synthetic Dataset** | Real certificate data is not public, so training was done on simulated samples. |
| **Only Structured Analysis** | The current model only checks structured fields (text values), not actual PDF or visual elements. |
| **No Document Image Input** | Does not analyze certificate layouts, logos, or digital seals. |
| **False Positives/Negatives** | If someone enters a real name with a tampered ID, the model might misclassify based on encoded features. |

**Suggestions to Improve or Scale the System:**

1. **PDF Metadata Analysis**
   * Integrate PyPDF2 or pdfplumber to check if the document’s creation and modification timestamps differ.
2. **OCR + Visual Layout Detection**
   * Use Tesseract OCR with OpenCV to extract text from certificate images and compare structure/font/spacing with a known valid template.
3. **Database/API Integration**
   * Connect to an official certificate validation API (if provided by Great Learning or similar platforms).
4. **Web Deployment**
   * Deploy the prototype as a web-based verification tool using Streamlit or Flask, allowing real-time validation with file upload.
5. **Real Dataset Inclusion**
   * Partner with credential providers to obtain anonymized, real-world certificates for training and benchmarking.

**Conclusion:**

The project successfully demonstrates a working prototype that can flag tampered academic credentials based on structured features. It provides a strong foundation for future enhancements using document forensics, metadata tracking, and visual pattern recognition.