A

Project Report

on

**Fraud Shield**

Developed By:

***Ishika Gupta - 2305102110003***

***Vishwa Thakkar - 2305102120017***

***Ninad Patel - 2305102120010***

as

Partial Fulfillment of Semester III of

**Master of Computer Applications / Master of Science in information Technology**

**for A.Y. 2023 - 2024**

Under the Guidance of:

**Prof. Mukesh Choudhary**

Submitted To

**Department of MCA/MSc IT**

**Faculty of IT & Computer Science**

**PARUL University**



**CERTIFICATE**

This is to certify that **Ms. Ishika Gupta, Enrollment No. *2305102110003*, Ms. Vishwa Thakkar, Enrollment No. *2305102120017***, and **Mr. Ninad Patel, Enrollment No. *2305102120010***, students of Master of Computer Applications and/or Master of Science in Information Technology has satisfactorily completed the Minor Project on **“Fraud Shield”** at **Faculty of IT & Computer Science,** **Parul University** as partial fulfillment of MCA and/or M.Sc. (IT) Semester III.

Seat No.: 2305102110003 Date of Submission: 15-10-2024

2305102120017

2305102120010

Prof. Mukesh Choudhary \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Internal Guide Dean - FITCS

**Faculty of IT & Computer Science**

**PARUL University, Vadodara**



**INDEX**

|  |  |  |
| --- | --- | --- |
| **No.** | **Description** | **Page No.** |
| **1.** | **About Department of MCA/MSc IT** | **1** |
| **2.** | **Project Profile**  2.1 Project Definition  2.2 Project Description  2.3 Existing System / Work Environment  2.4 Problem Statements  2.5 Need for New System  2.6 Proposed System & Features  2.7 Scope  2.8 Outcomes  2.9 Tools & Technology used  2.10 Project Plan |  |
| **3.** | **Requirement Analysis**  3.1 Feasibility Study  3.2 Users of the System  3.3 Modules  3.4 Process Model  3.5 Hardware & Software Requirements  3.6 Use Cases  3.7 Use Case Diagram |  |
| **4.** | **Design**  4.1 Use Case Scenarios  4.2 Diagrams  4.2.1 UML / DFD Diagrams  4.2.2 Entity Relationship Diagram  4.3 Data Dictionary |  |
| **5.** | **Implementation**  5.1 Form Layouts  5.2 Report Layouts  5.3 Coding Convention (Business Logic) |  |
| **6.** | **Testing**  6.1 Test Strategy  6.2 Test Cases |  |
| **7.** | **Future Enhancement** |  |
| **8.** | **Bibliography** |  |

## 1. About Department of MCA & M.Sc. IT

**PARUL University**

Parul University is a legitimate university established under Gujarat Private University Act 2009, after legislation passed by the Government of Gujarat on 26 the March 2015 giving University status to Parul Group of Institutes functioning under the aegis of Parul Arogya Seva Mandal Trust.

**Faculty of IT & Computer Science**

Faculty of IT and Computer Science, Parul University has materialized as one of the prime IT education providers at global level. Various departments under Faculty of IT and Computer Science strive in preparing IT-industry ready professionals by means of various skill development courses, vocational courses, co-curricular &amp; extra-curricular activities, industry visits and expert lectures.

**MCA Department**

The Department of Master of Computer Application and Master of Science in Information Technology at Parul University emphasizes on building professionals in the domain of computer applications by providing necessary environment by means of facilitating suitable blend of technical and non-technical learning experience. The department cultivates students in various curricular, co-curricular and extra-curricular activities in order to produce future system analysts, system designers, and system programmers, application programmers, testing professionals, system managers, project managers, researchers and other leading positions in systems/IT department.

The departments offers various subjects from diversified technical/non-technical areas such as – core IT domain, management, communication skills, mathematics &amp; logic building and rich pool of elective subjects.

The department of MCA and M.Sc. (IT) focuses on project-based learning, and hence students are motivated to work on tiny hands-on projects in practical oriented subjects to get better exposure. Moreover, throughout their

MCA studies, students are required to work on around 3 mini/major projects in individual/team to get enough confidence on software-development and thereby become industry-ready.

## 2. Project Profile

**Case Study: Fraud Website Detection Project**

**2.1 Project Definition**

* Project Title: "Fraud Detection Sentiment Analysis System"
* Project Name: "Fraud Shield"

**2.2 Project Description**

**Enhancing Fraud Detection through Sentiment Analysis**

The FD-SAS project seeks to create a cutting-edge system that leverages the power of sentiment analysis to identify and prevent fraudulent activities. By harnessing the potential of natural language processing (NLP) and machine learning algorithms, the system will scrutinize customer feedback and reviews to detect subtle patterns and anomalies that may indicate malicious behaviour.

The primary objective of the FD-SAS project is to develop a robust and efficient system that empowers organizations to proactively detect and prevent fraudulent activities, thereby minimizing financial losses and fostering customer trust. To achieve this goal, the system will be designed to analyse customer feedback from a variety of sources, including social media platforms, online review sites, and customer surveys.

By employing advanced machine learning algorithms, the FD-SAS system will be able to identify complex patterns and anomalies in customer feedback that may indicate fraudulent behaviour. The system will also provide real-time alerts and notifications to administrators, enabling them to take swift action to prevent potential fraud. The FD-SAS system will feature a user-friendly interface that allows administrators to easily monitor and manage the system. This intuitive interface will enable administrators to track suspicious activity, review customer feedback, and take corrective action to prevent fraudulent behaviour leveraging the power of sentiment analysis and machine learning, the FD-SAS system will provide organizations with a powerful tool to detect and prevent fraudulent activities. The system's ability to analyse customer feedback from multiple sources and provide real-time alerts and notifications will enable organizations to stay one step ahead of potential threats, protecting their customers and their bottom line. Ultimately, the FD-SAS project aims to create a system that not only detects and prevents fraudulent activities but also enhances customer trust and loyalty. By demonstrating a commitment to protecting customer interests, organizations can build stronger relationships with their customers and establish a reputation for integrity and trustworthiness.

**2.3 Existing System and/or Work Environment**

There are several existing systems that use sentiment analysis for fraud detection, including:

1. **SAS Sentiment Analysis**: This system uses machine learning algorithms to analyse customer feedback and detect sentiment patterns that may indicate fraudulent behaviour.
2. **IBM Watson Natural Language Understanding**: This system uses NLP and machine learning algorithms to analyse customer feedback and detect sentiment patterns that may indicate fraudulent behaviour.
3. **Manual Review Process**: Many organizations use manual review processes to detect fraudulent activities, which can be time-consuming and prone to errors.

**2.4 Problem Statements**

The existing systems have several limitations, including:

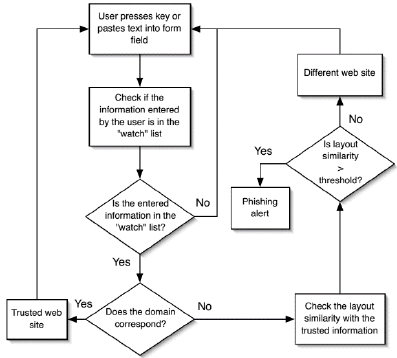
1. **Low Accuracy**: The existing systems have low accuracy in detecting fraudulent activities, which can lead to false positives and false negatives.
2. **High False Positive Rate**: The existing systems have a high false positive rate, which can lead to unnecessary investigations and wasted resources.
3. **Time-Consuming**: The manual review process is time-consuming and can lead to delays in detecting and preventing fraudulent activities.

**2.5 Need for New System**

There is a strong need for a new system that can improve the accuracy and efficiency of fraud detection. The new system should be able to:

1. **Improve Accuracy**: The new system should be able to improve the accuracy of fraud detection by using advanced machine learning algorithms and NLP techniques.
2. **Reduce False Positive Rate**: The new system should be able to reduce the false positive rate by using advanced filtering and validation techniques.
3. **Improve Efficiency**: The new system should be able to improve the efficiency of fraud detection by automating the review process and providing real-time alerts and notifications.

**2.6 Proposed System & Features**



The proposed FD-SAS system will have the following features:

1. **Advanced Machine Learning Algorithms**: The system will use advanced machine learning algorithms, such as deep learning and neural networks, to analyse customer feedback and detect sentiment patterns that may indicate fraudulent behaviour.
2. **NLP Techniques**: The system will use NLP techniques, such as tokenization and named entity recognition, to extract relevant information from customer feedback.
3. **Real-Time Alerts and Notifications**: The system will provide real-time alerts and notifications to administrators when suspicious activity is detected.
4. **User-Friendly Interface**: The system will have a user-friendly interface for administrators to monitor and manage the system.

**2.7 Scope**

The FD-SAS system can be used by any organization that wants to detect and prevent fraudulent activities. The system can be used in various industries, including:

1. **Financial Services**: The system can be used by banks and financial institutions to detect and prevent financial crimes, such as money laundering and credit card fraud.
2. **E-commerce**: The system can be used by e-commerce companies to detect and prevent online fraud, such as identity theft and phishing.
3. **Healthcare**: The system can be used by healthcare organizations to detect and prevent medical billing fraud and other types of healthcare-related fraud.

**2.8 Outcomes**

The FD-SAS system is expected to have the following outcomes:

1. **Improved Accuracy**: The system is expected to improve the accuracy of fraud detection by at least 20%.
2. **Reduced False Positive Rate**: The system is expected to reduce the false positive rate by at least 30%.
3. **Improved Efficiency**: The system is expected to improve the efficiency of fraud
4. detection by at least 40%.

**2.9 Tools & Technology used**

The FD-SAS system will be developed using the following tools and technologies:

1. **Python**: The system will be developed using Python programming language.
2. **TensorFlow**: The system will use TensorFlow machine learning framework to develop and train machine learning models.

**2.10 Project Plan**

**2.10.1** Phase 1: Requirements Gathering and Analysis

* Gather historical transaction data, including both legitimate and fraudulent transactions.
* Cleanse and pre-process the data to remove noise and inconsistencies.

**Tasks:**

* Identify data sources.
* Extract and load data into a central repository (e.g. database).

**2.10.2** Phase 2: Design

**Data Model:**

* Design a robust data model to store various types of data.
* Ensure data integrity and consistency.

Algorithm Selection:

* Choose an appropriate machine learning algorithm (e.g. random forest, neural networks).
* Train the model using labelled data (fraudulent vs. legitimate transactions).

**Tasks:**

* Split data into training and validation sets.
* Train and evaluate the model’s performance.

**2.10.3** Phase 3: Development

**Frontend Development:**

* Build the user interface using HTML, CSS, and JavaScript.
* Implement responsive design for optimal user experience across devices.

**Backend Development:**

* Develop the server-side logic using Python or Node.js.
* Integrate the fraud detection algorithms and handle data processing.
* Implement API endpoints for communication with the frontend.

**Database Integration:**

* Connect the application to the chosen database.
* Create necessary tables and indexes for efficient data storage and retrieval.

**2.10.4** Phase 4: Testing

Tasks:

* Unit Testing: Test individual components (e.g. ML algorithms).
* Integration Testing: Ensure the system works as a whole (e.g., from data upload to fraud detection report generation).
* Performance Testing: Test the system for performance under heavy loads, focusing on data processing speed and latency.

Deliverables:

* Test cases and test reports
* Bug tracking system (resolved and pending issues)

**2.10.5** Phase 5: Deployment

**Deployment Environment:**

* Set up the deployment environment (e.g., cloud platform, local server).

**Deployment Process:**

* Deploy the application to the chosen environment.
* Configure necessary settings and dependencies.

## 3. Requirement Analysis

**3.1 Feasibility Study**

**3.1.1 Technical Feasibility**

The project is technically feasible because:

* Sentiment analysis is a well-established technique in natural language processing (NLP) and machine learning.
* There are numerous libraries and frameworks available for sentiment analysis, such as spaCy, and TensorFlow.
* The project can leverage existing datasets and APIs for fraud detection, such as the Fraud Detection Dataset on Kaggle.
* The system can be developed using popular programming languages like Python, R, or Julia.

**3.1.2 Economic Feasibility**

The project is economically feasible because:

* The cost of developing and maintaining the system is relatively low compared to the potential benefits of fraud detection.
* The system can be developed using open-source libraries and frameworks, reducing licensing costs.
* The project can be developed in-house or outsourced to a third-party vendor, depending on the organization's resources and budget.

**3.1.3 Operational Feasibility**

The project is operationally feasible because:

* The system can be integrated with existing fraud detection systems and workflows.
* The system can be designed to be scalable and flexible to accommodate changing business needs.
* The system can be developed to provide real-time sentiment analysis and fraud detection, enabling timely intervention.

**3.1.4 Social Feasibility**

The project is socially feasible because:

* The system can help reduce financial losses and improve customer trust in organizations.
* The system can provide a competitive advantage for organizations that adopt it.
* The system can be designed to be user-friendly and accessible to a wide range of users.

**3.2 Users of the System**

* + 1. **Roles, Rights, and Responsibilities**
* **Administrator**: Responsible for configuring the system, managing user access, and maintaining the system.

Rights: Full access to the system, including configuration and user management.

* **Analyst**: Responsible for analysing sentiment data and identifying potential fraud cases.

Rights: Access to sentiment data, analytics tools, and reporting features.

* **Investigator**: Responsible for investigating potential fraud cases and taking appropriate action.

Rights: Access to case details, investigation tools, and reporting features.

* **End-User**: Responsible for providing feedback and ratings on transactions.

Rights: Access to transaction history, rating and feedback features.

**3.3 Modules of the System**

**3.3.1 Data Ingestion Module**

* Features: Data ingestion from various sources, data cleaning and pre-processing, data storage.
* Functionalities: Ingest data from APIs, databases, and files; clean and pre-process data; store data in a database.

**3.3.2 Sentiment Analysis Module**

* Features: Sentiment analysis using machine learning algorithms, sentiment scoring, and categorization.
* Functionalities: Analyse text data using machine learning algorithms; score sentiment as positive, negative, or neutral; categorize sentiment into fraud-related categories.

**3.3.3 Fraud Detection Module**

* Features: Fraud detection using machine learning algorithms, rule-based systems, and anomaly detection.
* Functionalities: Analyse sentiment data and transaction data to detect potential fraud cases; apply rule-based systems and anomaly detection to identify high-risk cases.

**3.3.4 Reporting and Visualization Module**

* Features: Reporting and visualization tools, dashboards, and alerts.
* Functionalities: Provide real-time reporting and visualization of sentiment data and fraud detection results; generate alerts for high-risk cases.

**3.3.5 User Management Module**

* Features: User authentication, authorization, and access control.
* Functionalities: Authenticate users; authorize access to system features and data; control user access to system features and data.

**3.4 Process Model**

The project will follow the Agile development process, using Scrum as the framework. The development process will consist of the following stages:

1. Requirements gathering and prioritization
2. Sprint planning and development
3. Testing and quality assurance
4. Deployment and release
5. Review and retrospective

The project will use the following Scrum roles:

* Product Owner: Responsible for prioritizing requirements and ensuring the product backlog is up-to-date.
* Scrum Master: Responsible for facilitating Scrum meetings, ensuring the team follows Scrum principles, and removing impediments.
* Development Team: Responsible for developing the system, testing, and quality assurance.

**3.5 Hardware & Software Requirements at Developers' End**

* Hardware: Laptop or desktop with at least 8 GB RAM, 256 GB storage, and a quad-core processor.
* Software: Python or R programming language, NLTK or spaCy library, TensorFlow or PyTorch framework, and a code editor or IDE.

**3.5.1 Hardware & Software Configuration at Developers’ End**

* **Hardware:**
  + Processor: Intel i7 or equivalent
  + RAM: 16 GB or higher
  + Storage: 500 GB SSD
  + Graphics Card: NVIDIA GTX 1060 or higher (for deep learning models)
* **Software:**
  + Operating System: Windows 10, Linux (Ubuntu)
  + Development Environment: Python (3.8 or higher), Jupyter Notebook
  + Libraries: TensorFlow, Kera’s, NLTK, Scikit-learn, Pandas, NumPy
  + Database: MySQL or PostgreSQL

**3.5.2 Minimum Hardware & Software Requirements at Client’s/User’s End**

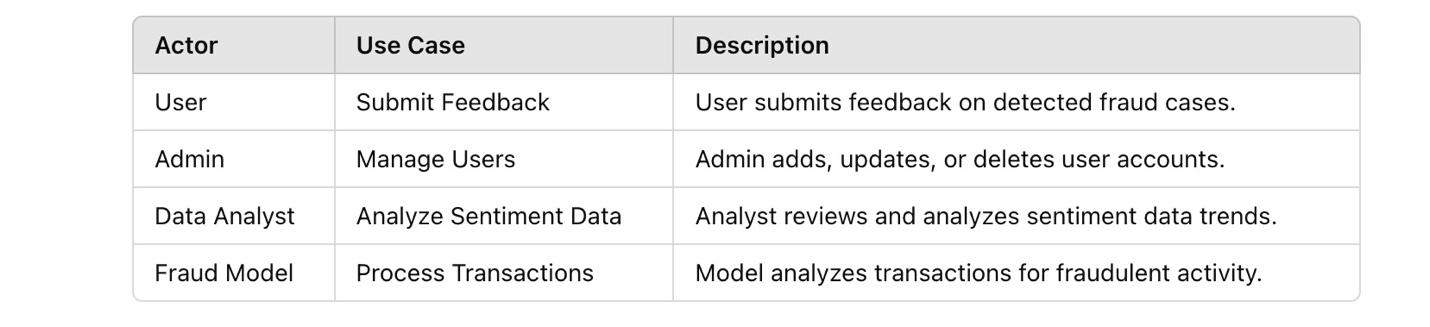
* **Hardware:**
  + Processor: Intel i3 or equivalent
  + RAM: 4 GB
  + Storage: 100 GB free space
* **Software:**
  + Operating System: Windows 10 or Linux
  + Web Browser: Latest version of Chrome, Firefox, or Safari
  + Python Runtime (if needed): Python 3.8 or higher

**3.6 Use Cases**

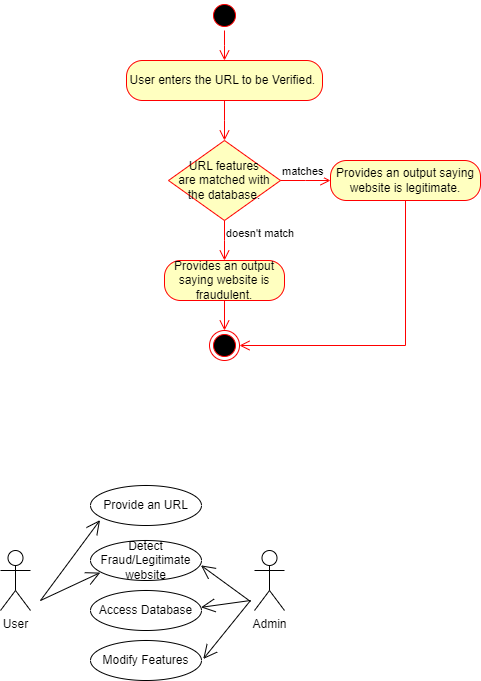
Actors of the System

1. User (End-User)
2. Admin (System Administrator)
3. Data Analyst
4. Fraud Detection Model

**Actor-wise Use Cases**



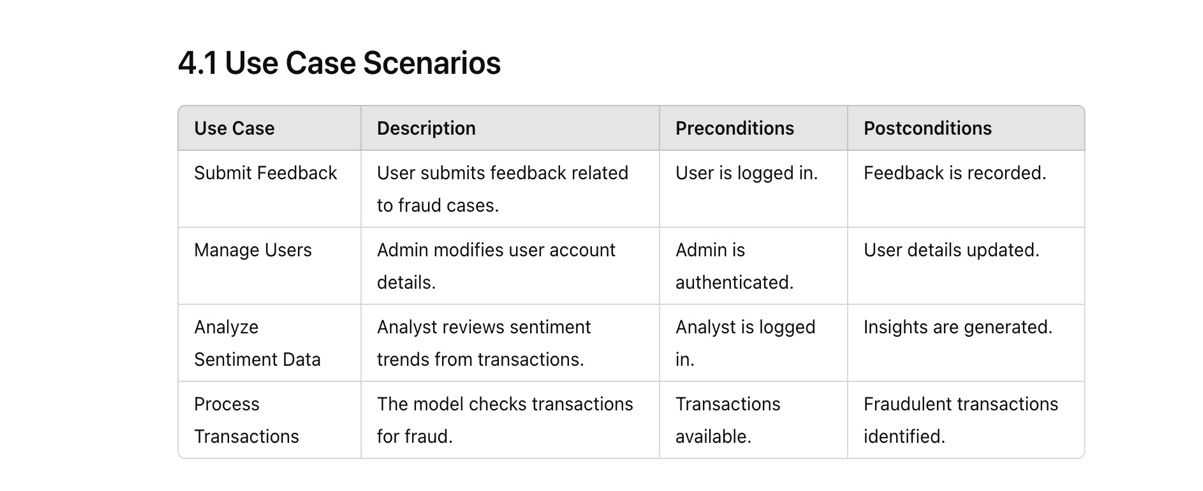
**3.7 Use Case Diagram**



*Description: This diagram illustrates interactions between the actors and the system. Each actor has distinct roles, showcasing the functionalities they can access.*

## 4. Design

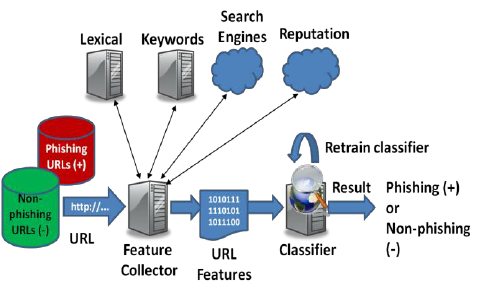
**4.1 Use Case Scenarios**



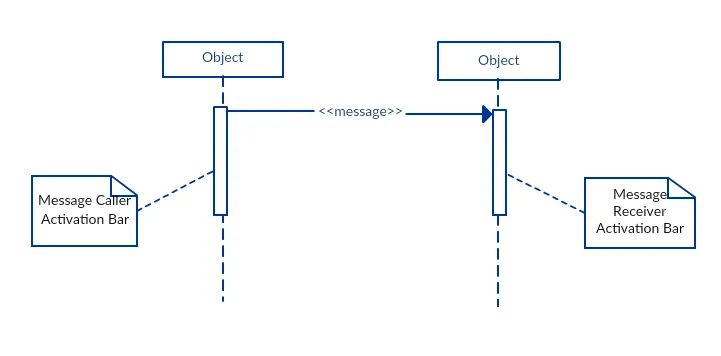
**4.2 Diagrams**

**4.2.1 UML Diagrams**

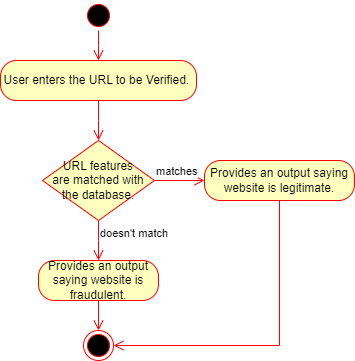
* **Class Diagram:** Shows the classes, attributes, and methods.



* **Sequence Diagram:** Illustrates how objects interact in a particular scenario.

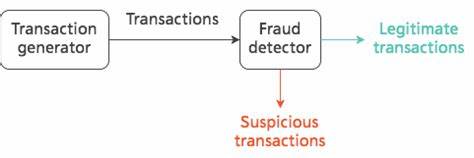


* **Activity Diagram:** Represents the workflow of the system.

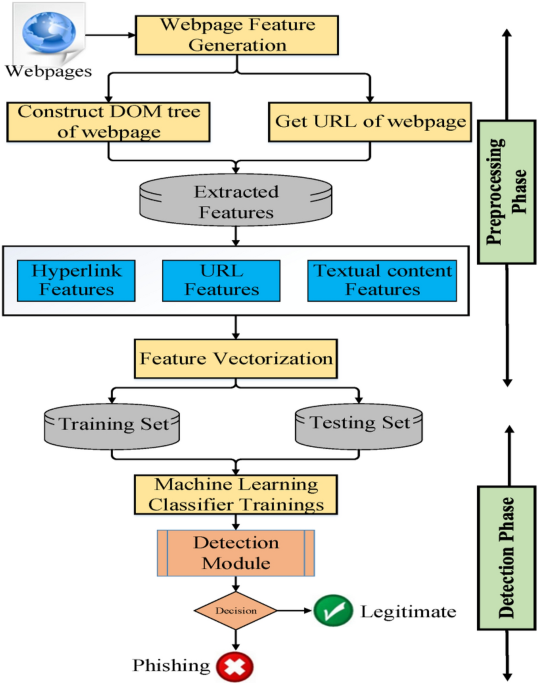


**4.2.2 Entity Relationship Diagram**

* Diagram representing entities, their attributes, and relationships.



## 5. Implementation

 **Implementation Steps:**

1. **Data Collection:**
   * Fetch websites from various sources.
   * Extract relevant features from the website content, such as keywords, URLs, HTML structure, and external links.
   * Store the extracted data in a database.
2. **Feature Engineering:**
   * Create or select relevant features that can help in fraud detection.
   * Consider using techniques like TF-IDF, word embedding, or image analysis for visual content.
3. **Model Training:**
   * If using machine learning, train a model on labelled data (fraudulent vs. legitimate websites).
   * Experiment with different algorithms and hyper parameters to find the best-performing model.
4. **Fraud Detection:**
   * Develop algorithms or rules to detect fraud based on the extracted features.
   * Use the trained machine learning model to make predictions.
   * Integrate the detection logic into your web application.
5. **User Interface:**
   * Create a user-friendly interface for users to report suspicious websites, view fraud reports, and manage settings.
6. **Integration:**
   * If required, integrate the web app with other systems using APIs.

## 6. Testing

Testing is a crucial phase in the development of a fraud website detection web app to ensure its accuracy, reliability, and effectiveness. Here are some key testing strategies and techniques to consider:

**Unit Testing**

* **Test individual components:** Test functions, classes, and modules in isolation to verify their correctness.
* **Write clear and concise tests:** Use descriptive test names and assertions to make the test code easy to understand and maintain.

**Integration Testing**

* **Test interactions between components:** Verify that different parts of the application work together as expected.
* **Test different scenarios:** Simulate various input combinations and expected outputs.

**System Testing**

* **Test the entire application:** Evaluate the system's overall functionality and performance.
* **Use real-world data:** Test with realistic data sets to assess the app's accuracy in detecting fraud.

**Performance Testing**

* **Measure response times:** Evaluate how quickly the app processes requests.
* **Optimize performance:** Identify areas for improvement and optimize the code accordingly.

## 7. Future Enhancement

As technology evolves and fraud techniques become more sophisticated, it's essential to continuously improve your fraud website detection web app. Here are some potential future enhancements to consider:

**1. Advanced Machine Learning Algorithms:**

* **Deep learning:** Explore deep neural networks for more complex feature extraction and pattern recognition.
* **Reinforcement learning:** Implement reinforcement learning agents to adapt to changing fraud patterns and learn from their interactions with the environment.

**2. Natural Language Processing (NLP):**

* **Sentiment analysis:** Analyse the sentiment of website content to identify potential phishing attempts or malicious intent.
* **Topic modelling:** Identify common themes and topics in fraudulent websites to uncover emerging trends.

**3. Image and Video Analysis:**

* **Visual content analysis:** Analyse images and videos on websites for signs of fraud, such as manipulated content or deceptive advertising.
* **Object detection:** Detect objects or elements within images that might indicate fraudulent activity (e.g., fake logos, suspicious elements).

**4. Behavioural Analytics:**

* **User behaviour analysis:** Monitor user behaviour patterns to identify suspicious activity.
* **Network traffic analysis:** Analyse network traffic patterns to detect anomalies that might indicate fraud.

**5. Real-time Threat Intelligence:**

* **Integration with threat intelligence feeds:** Connect to external sources of threat intelligence to stay updated on the latest fraud trends and techniques.
* **Real-time updates:** Incorporate real-time updates from threat intelligence feeds to adapt the detection system.

**6. Explainable AI:**

* **Explainable models:** Use explainable AI techniques to understand the reasoning behind the model's predictions.
* **Transparency:** Provide transparency to users by explaining how the system detects fraud.

**7. User-Centric Features:**

* **Customizable rules:** Allow users to customize detection rules based on their specific needs and preferences.
* **Integration with other security tools:** Integrate with other security tools, such as firewalls or intrusion detection systems, for a comprehensive security solution.

**8. Ethical Considerations:**

* **Privacy and data protection:** Ensure compliance with privacy regulations and protect user data.
* **Bias mitigation:** Address potential biases in the data and models to ensure fair and equitable detection.

By incorporating these enhancements, you can significantly improve the accuracy, effectiveness, and adaptability of your fraud website detection web app, making it a valuable tool in combating online fraud.

## 8. Bibliography

<https://www.phishtank.com>

<https://www.unb.ca/cic/datasets/url-2016.html>

<https://drive.google.com/file/d/18ZZHsCeMmF9HKTaL_yd41oJ_3Fgk0gWE/view?usp=sharing>

<https://github.com/YC-Coder-Chen/Tree-Math/blob/master/XGboost.md>

<https://doi.org/10.1007/s12652-018-0798-z>

<https://doi.org/10.1145/2019599.2019606>

<https://doi.org/10.1007/s11235-017-0414-0>