Team : Taggers

Problem statement 1 :

There are social engineering frauds in which fake accounts of prominent people are created on social media and their friends are approached to transfer certain money into the accounts of fraudsters. Give a technical solution to be adopted by social media firms in which such types of fraud accounts are automatically deleted.

Solution for the problem :

Social media fraud involving fake accounts impersonating prominent individuals is a growing cybersecurity threat. Fraudsters create duplicate accounts, use similar profile pictures and bios, and trick victims into financial scams. The proposed solution uses blockchain and AI to automatically verify and authenticate accounts, preventing fake ones from being created in the first place.

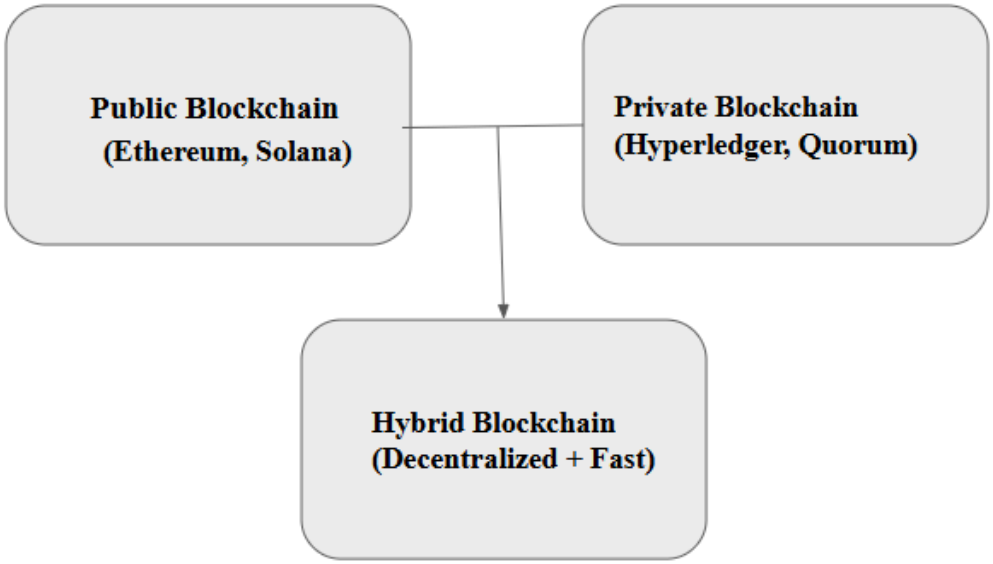
- ❖ **Key Features of the Solution**
  - **Blockchain-based digital identity (DID) for verified users**
  - **AI-powered similarity detection (profile images, names, bios)**
  - **Automated fraud prevention and blacklisting of fake accounts**
  - **Decentralized and tamper-proof identity registry**

Stepwise explanation of solution:

Step 1: Selecting the Blockchain Infrastructure

- Choosing the right blockchain setup (Public, Private, or Hybrid).

➔ **Hybrid blockchain is best for balance between security & speed.**

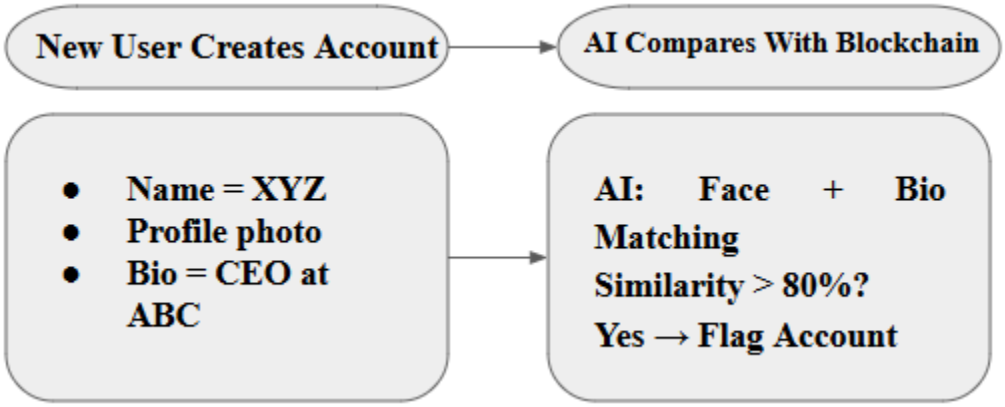


Step 2: Creating Blockchain-Based Digital Identities (DID)

- Every verified account is assigned a unique digital identity on the blockchain.

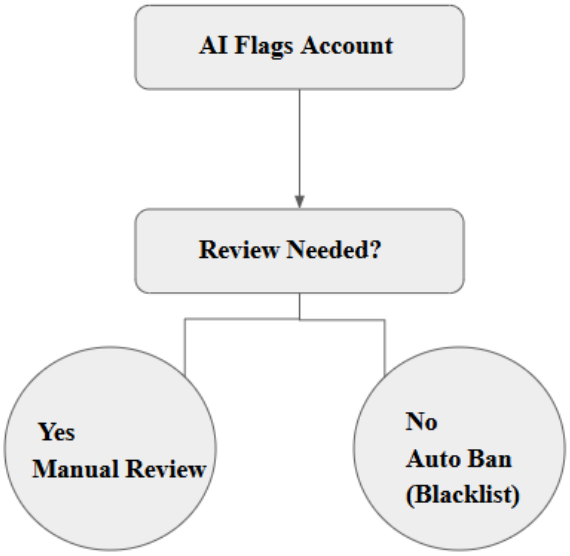
Step 3: AI-Powered Fake Account Detection Before Activation

- System checks profile similarity before allowing activation.



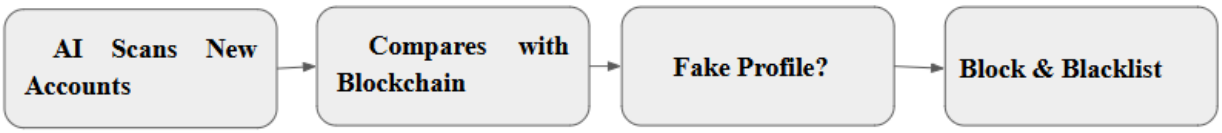
Step 4: Flagging and Blocking Fraudulent Accounts

- If a fake account is detected, it's flagged or blocked before activation.



Step 5: Continuous Monitoring and Updates

- AI & Blockchain ensure ongoing fraud detection.



Implementation Steps:

Step 1: Select a Blockchain Infrastructure

Choose a blockchain network based on the platform's scalability and security needs. Options include:

- **Public Blockchain** (e.g., Ethereum, Solana) – Highly decentralized but costly for frequent transactions.
- **Private Blockchain** (e.g., Hyperledger Fabric) – Better for enterprise-level identity verification.
- **Hybrid Blockchain** – Combines features of both for balance between security and efficiency.

Step 2: Generate a Digital Identity for Verified Users

Each high-profile user undergoes identity verification (KYC) and is assigned a **unique digital identity (DID)** stored on the blockchain.

1. **User submits identity proof (e.g., passport, government ID).**
2. **A cryptographic hash** of the user's name, photo, and ID details is generated.
3. **This hash is stored on the blockchain** as a non-fungible token (NFT) or smart contract entry.
4. **A verification badge** is issued to the user, proving their authenticity.

Tech Stack:

- **Decentralized Identifiers (DIDs) Protocol**
- **Smart Contracts (Solidity, Rust, or Chaincode for Hyperledger)**
- **IPFS (InterPlanetary File System) for Storing Profile Pictures Securely**

Step 3: Develop an AI-Based Detection System

To check for impersonation, every new account undergoes **AI-powered identity verification** before activation.

1. **When a new account is created, extract its name, bio, and profile picture.**
2. **AI compares these details with blockchain-verified identities** using:
    - **Facial recognition** (CNN-based models like FaceNet or OpenCV).
    - **Text similarity analysis** (NLP-based embeddings like BERT or TF-IDF).
3. **If similarity exceeds a threshold (e.g., 80%), the account is flagged.**

Tech Stack:

- **OpenCV, FaceNet** (for facial similarity checks)
- **BERT, TensorFlow/Keras** (for NLP-based bio verification)
- **Smart Contracts + Chain Link Oracles** (for blockchain-data retrieval)

Step 4: Flag or Block Fraudulent Accounts

1. If a new account is flagged for similarity:
    - **System notifies the user for manual verification.**
    - **Asks for additional ID proof** (handled via blockchain smart contracts).
2. If fraud is confirmed:
    - **Account is automatically rejected and blacklisted.**
    - **Metadata is added to the fraud database** to prevent repeat attempts.
3. **If an error occurs (false positive), an appeal mechanism is available.**

Tech Stack:

- **AI Fraud Detection Module** (Python/Node.js)
- **Smart Contract Updates on Blockchain**

Step 5: Continuous Monitoring and Updates

1. AI continuously scans **existing and new accounts** for impersonation.
2. **Regular updates to blockchain identity registry** when a verified user updates profile details.
3. **Periodic audits** via smart contracts ensure tamper-proof identity tracking.

Tech Stack:

- **Automated AI APIs** (for monitoring new accounts)
- **Blockchain API Integration** (via Ethereum/Solana RPCs)

Test Cases:

- **Scenario 1: Fake Account Attempt (Impersonating "John Doe")**

Input:

```
name = "John Doe"
bio = "CEO at XYZ Corp"
profile_pic = "fake_john.jpg"
verified_img = "real_john.jpg"
result = detect_fake_account(name, bio, profile_pic, verified_img)
print(result)
```

Expected Output:

Fake Account Alert! Profile picture is too similar to a verified user.

Why?

➔ **Facial Recognition detects similarity with a real user.**
➔ **Blockchain confirms an existing identity.**

- **Scenario 2: Fraudulent Bio (Using Similar Bio of a Verified Person)**

Input:

```
name = "Random User"
bio = "CEO at XYZ Corp"
profile_pic = "random_person.jpg"
verified_img = "real_john.jpg"
result = detect_fake_account(name, bio, profile_pic, verified_img)
print(result)
```

Expected Output:

Suspicious Account! Bio is too similar to a verified user.

Why?

- **Bio similarity detected using NLP** (Text embeddings found **80%+ similarity**).
- **Profile pictures might be different, but impersonation still exists.**
- **Scenario 3: Genuine User (No Fraud Detected)**

Input:

```
name = "Mike Johnson"
bio = "Software Engineer at Google"
profile_pic = "mike.jpg"
verified_img = "real_john.jpg"
result = detect_fake_account(name, bio, profile_pic, verified_img)
print(result)
```

Expected Output:

```
Account Approved! No fraud detected.
```

Why?

➔ **No similar name, bio, or profile picture found in the blockchain.**
➔ **AI confirms that the user is original.**

- **Scenario 4: Direct Blockchain Match (Existing User Trying to Register Again)**

Input:

```
name = "Alice Smith"

bio = "Influencer at XYZ"

profile_pic = "alice_verified.jpg"

result = check_blockchain_identity(name, bio, profile_pic)
print(" Fake Account Detected!" if result else "Account Approved!")
```

Expected Output:

```
Fake Account Detected!
```

Conclusion: Why Does This Work?

- **Tamper-Proof**: Blockchain records prevent unauthorized identity changes.
- **AI-Powered Detection**: Real-time fraud prevention using deep learning.
- **Seamless User Experience**: Verified users remain unaffected, while fraudsters get blocked.

This system can revolutionize **social media fraud detection** .

This shows how blockchain + AI can prevent fake accounts before activation, making social media safer.