FINANCE LOG ANALYSER

INNOVATE-X

Team AlgoAllies

ALGO ALLIES

Our Team

Ayush Arora - ayush_arora.ug22@nsut.ac.in

Meet Singh - meet.singh.ug22@nsut.ac.in

Krish Gupta - krish.gupta.ug22@nsut.ac.in

Manas Madan - manas.madan.ug22@nsut.ac.in

Tanish Saxena - tanish.saxena.ug22@nsut.ac.in

Problem Statement

- Financial institutions face increasing fraudulent activities, resulting in significant monetary losses and eroding customer trust, with traditional detection methods proving less effective against sophisticated fraud techniques.

- There's a critical need for a real-time fraud detection system that leverages machine learning to identify anomalies and fraud patterns while minimizing false positives and efficiently managing fraud cases.

Solution

- **FRAUD DETECTION SYSTEM:** We have developed a comprehensive fraud detection system that integrates machine learning, real-time data processing, and user-friendly interfaces for effective identification and management of fraudulent activities in financial transactions.

The system includes:

- Machine Learning Models for anomaly detection and identifying fraud patterns.
- Backend Services for processing and analyzing transaction data in real time.
- User Interface for monitoring transactions and receiving alerts.
- Case Management Features for investigating and managing fraud cases.

Solution

Key Features:

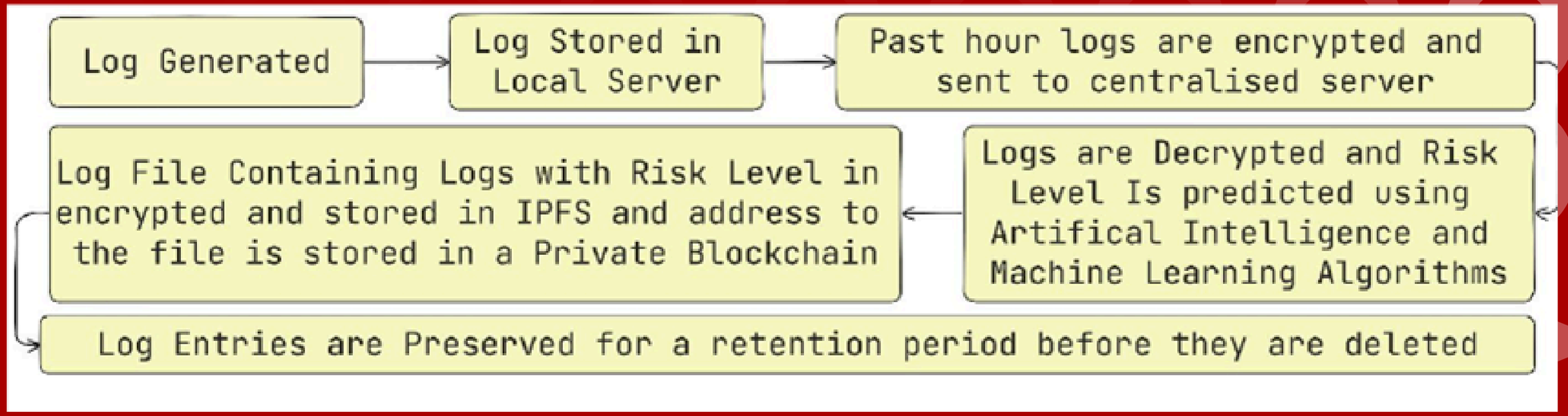
- Real-Time Fraud Detection: Utilizing advanced machine learning models to identify suspicious activities as they occur.
- User Interface: A dedicated interface for users to monitor transactions, receive alerts, and manage cases.
- Robust Data Processing: Backend services designed to handle and analyze large volumes of transaction data swiftly and accurately.
- Investigation and Management: Tools and features that aid in the thorough investigation and management of detected fraud cases, ensuring efficient resolution.

This system ensures a secure, reliable, and user-friendly approach to managing and mitigating fraudulent activities in financial environments.

USP

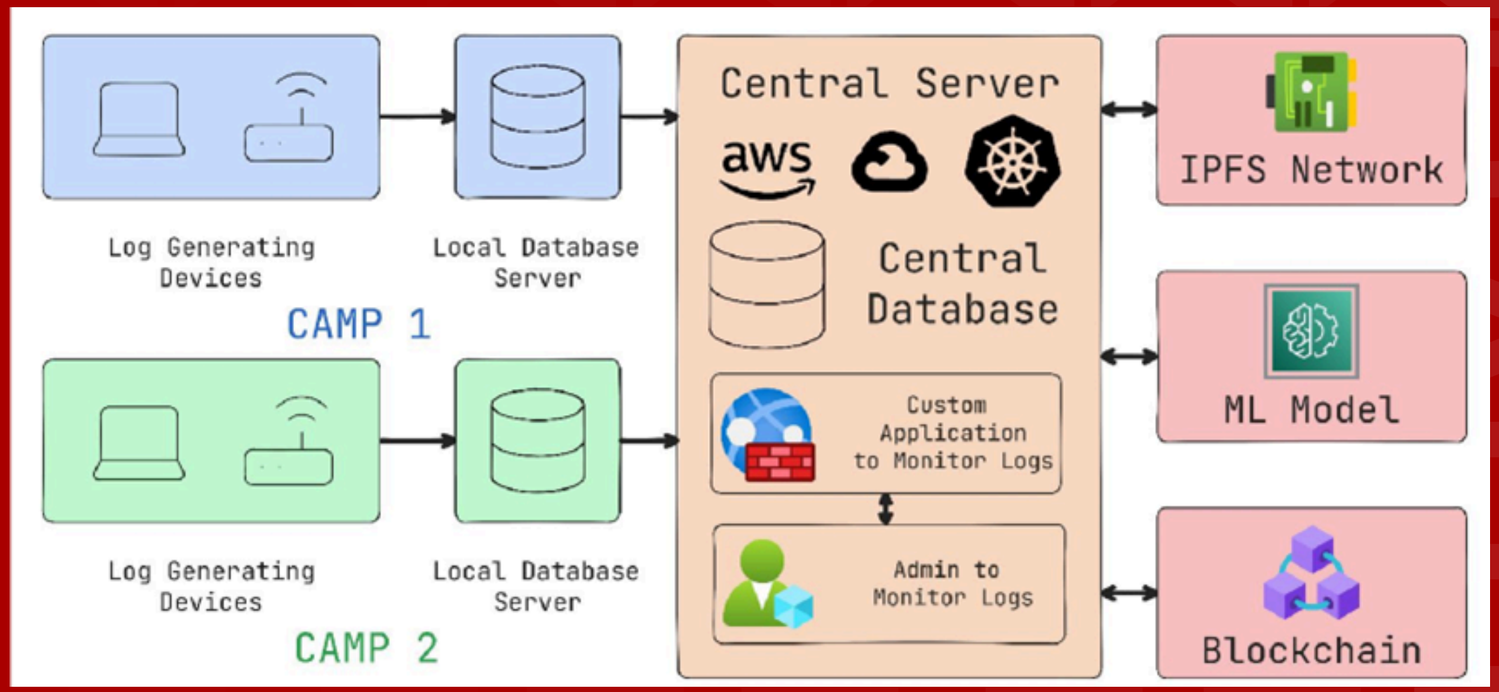
- Our system utilizes cutting-edge machine learning algorithms to detect anomalies and identify fraud patterns, ensuring accurate and efficient fraud detection.
- With robust backend services, our system processes and analyzes transaction data in real time, enabling instant identification and response to fraudulent activities.
- The intuitive interface allows users to monitor transactions, receive alerts, and navigate the system easily, enhancing user experience and operational efficiency.
- Efficient tools for investigating and managing fraud cases.

Flow Chart


Log Generated
Log Stored in Local Server
Past hour logs are encrypted and sent to centralised server
Logs are Decrypted and Risk Level Is predicted using Artifical Intelligence and Machine Learning Algorithms
Log File Containing Logs with Risk Level in encrypted and stored in IPFS and address to the file is stored in a Private Blockchain
Log Entries are Preserved for a retention period before they are deleted

Tech Stack

Infra (Diagram)



Why Not Just Blockchain

- Storing data on a blockchain can be expensive, as each node in the network must replicate and maintain a copy of the entire blockchain, including the stored data. This can lead to high storage costs, especially for large datasets.

- As the number of transactions and data stored on the blockchain increases, the network can become congested, leading to slower transaction processing times and higher fees.

- Executing operations on a blockchain, such as storing data or executing smart contracts, incurs gas costs.

What is IPFS

- IPFS is a open sourced modular suite of protocols for organizing and transferring data, designed from the ground up with the principles of content addressing and peer-to-peer networking.
- IPFS works by connecting computers in a network to enable them to share files. Each file and its contents are given a unique hash, and when someone requests a file, the network locates it based on that hash.
- Video on how data is stored in IPFS
- Idea behind IPFS

Why IPFS

- IPFS uses cryptographic hashes to uniquely identify content based on its actual data. This ensures that the address of a file is determined by its content, making it **tamper-evident** and **immutable.**

- IPFS provides faster access to data by enabling it to be replicated to and retrieved from multiple locations, and allowing users to access data from the nearest location using content addressing instead of location-based addressing.

- IPFS is an open, distributed and participatory network that reduces data silos from centralized servers, making IPFS more resilient than traditional systems

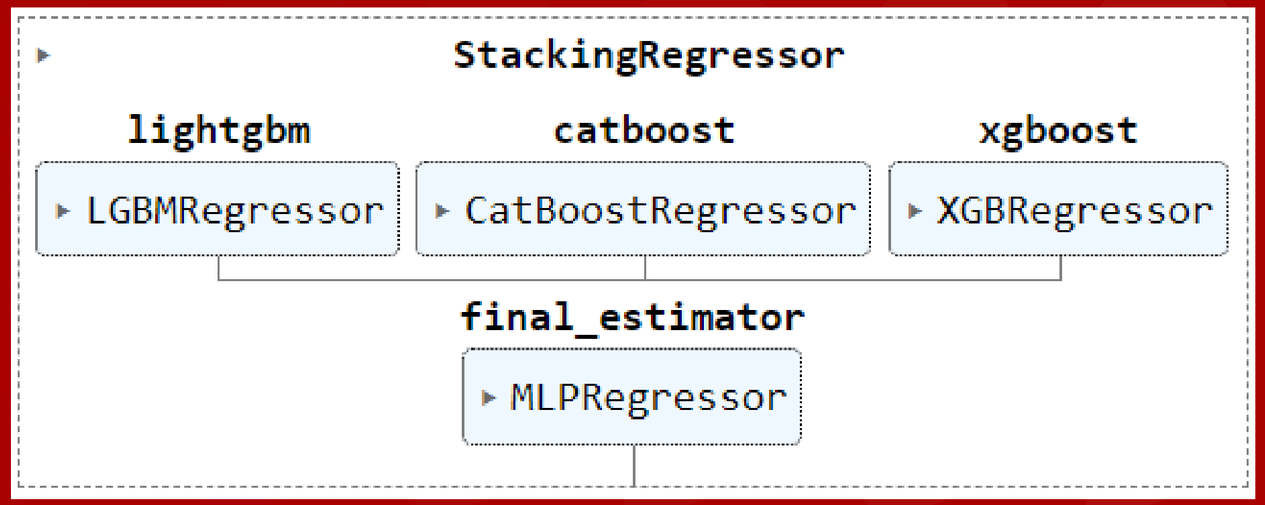
Why IPFS

- Failure of a single or even multiple nodes in the network does not affect the functioning of the entire network.
- IPFS is an open, distributed and participatory network that reduces data silos from centralized servers, making IPFS more resilient than traditional systems.
- Once a file is added to the IPFS network, the content of that file cannot be changed without altering the content identifier (CID) of the file. This feature is excellent for storing data that does not need to change .
- IPFS prevents vendor lock-in

ML Model

- An ML model is integrated into the project to predict the **risk percentage of entered logs.**
- The model is currently trained on a dummy dataset, and it will remain fully functional when replaced with a real dataset with the same features.
- We've compiled a dummy dataset with various features extracted from the logs.
- This is a **regression model** predicting a continuous output from 0 to 1 (1 being most risky, 0 being least) based on input features.

ML Model



- In this model, we employ the Stacking concept, combining three strong regressors - **XGBoost, CatBoost, and LightGBM.**
- A meta regressor, **MLPRegressor**, is used to finalize the model.
- Instead of a standard train-test split, we opt for **cross-validation** to mitigate overfitting risks.

ML Model

- Since we are using a dummy dataset it is difficult to predict how well the model will be in practice, but with techniques like stacking and cross validation we end up with a model that is **very well generalized.**

- With access to a real dataset, we can try **hyperparameter tuning, using different models, and feature engineering** to further improve our model.

Security Concerns

- If a gateway provider wants to limit access to requests with authentication, they may need to configure a reverse proxy, develop an IPFS plugin, or set a cache layer above IPFS.

- Configuring a reverse proxy is the most popular way for providers to handle authentication. Reverse proxy can also keep the original IPFS API calls which makes the gateway adaptable to all IPFS SDK and toolkits.

Security Concerns

- When sensitive files are stored using IPFS, they are broken into smaller pieces and distributed across multiple nodes. Retrieving the complete file requires consensus from most nodes, making it inherently more secure and resilient.

- We can store hashed references to sensitive data on the blockchain, ensuring that any changes to the data are immediately evident.

- Smart contracts can be programmed to execute access control logic and encrypting the data stored on IPFS ensuring that only parties with the appropriate permissions can retrieve or modify specific data stored on IPFS.

Architecture Safe from Hackers?

- The consensus mechanism plays a vital role in the security of a blockchain. Select a proven and secure consensus algorithm, such as Proof of Work (PoW) or Proof of Stake (PoS).
- Implement firewalls to control incoming and outgoing traffic, and use network segmentation to isolate different components of our blockchain network.
- IPFS has strict set of rules that does not allow unauthorised access of logs.
- Our System being accessible only to a private network logs can only be accessed by just authorised systems.

Architecture Safe from Hackers?

- We implemented Strong authentication mechanisms, such as multi-factor authentication (MFA), for accessing the systems
- Implement network segmentation to isolate the private network from external threats
- Configure firewalls to allow only necessary traffic to and from authorized systems. Restrict access to specific IP addresses and ports.
- Use Transit Encryption
- Make sure Employee Training is given priority.

What to do with Trillions of Logs?

- Periodically review and prune logs that are no longer essential for the integrity of the blockchain.
- Archived data can be stored more cost-effectively, while critical information remains on the blockchain.
- Larger datasets, represented by extensive logs, enable the application of advanced analytics and machine learning algorithms for predictive modeling.

What to do with Trillions of Logs?

- More logs provide a rich source of data for research, analysis, and optimization. Researchers and analysts can study historical patterns, trends, and user behaviors to derive insights.

- Storing logs on IPFS in expensive as duplication of file happens. We can delete logs from IPFS and store them in a traditional manner using compression techniques for efficient storage after some time

Minimizing Cloud Dependency

- We Plan to reduce the dependency on Cloud Providers like AWS by using On-Premises Data Center for storing data as they are cheaper to maintain and build.

- We can also take advantage of current infrastructure by adapting to a Hybrid or Private Cloud Model

Thank You !