Fraud Detection Pipeline - Case Study Review and Proposed Architecture

1. Introduction

In this document, I review a set of real-world case studies on fraud detection systems that showcase varied architectures, pipelines, and performance outcomes. Each case's pipeline is summarized, followed by key conclusions. Based on these learnings, I propose my own fraud detection pipeline—comparable to my Terraform-based AWS architecture.

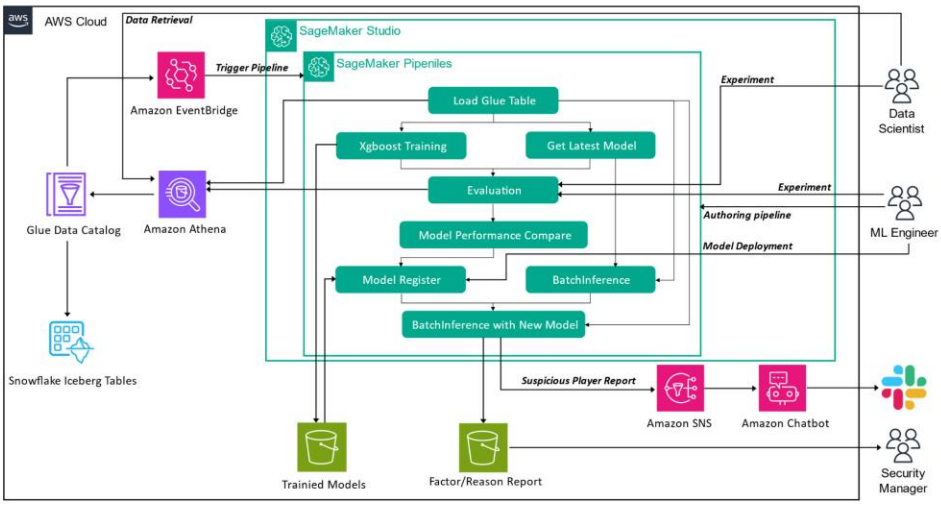
2. Case Study Reviews

A. NSUS Group – Gaming Industry Fraud Prevention (2025)

Link : [NSUS Group Enhances Financial Fraud Prevention System with Amazon SageMaker]

Pipeline & Architecture:
- Data ingestion into Amazon S3, creating a secure scalable data lake.
- ETL/data integration using AWS Glue with Apache Iceberg.
- Model development, hyperparameter tuning, and deployment via Amazon SageMaker Pipelines.
- Real-time inference and batch API integrations for detection.
- Storage of artifacts back onto S3.

Conclusions:
- Achieved AUC of ~95%, 30% greater business efficiency, and reduced manual efforts by over 80%.
- SageMaker Pipelines enabled rapid, minimal-human deployments and optimized features (270 → 80) for efficiency.

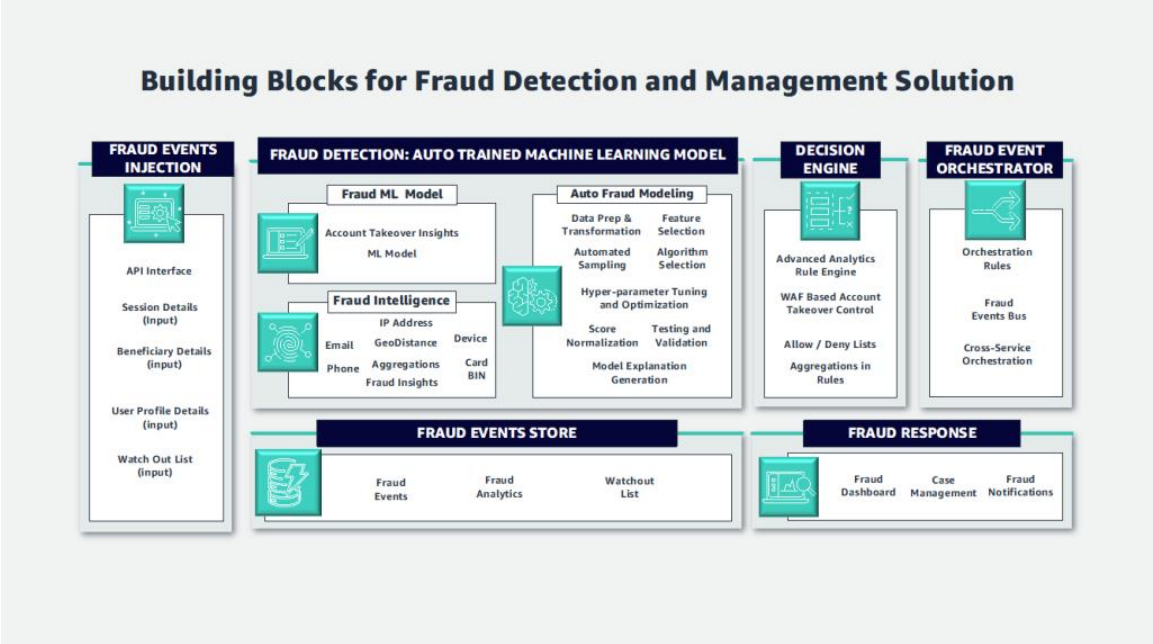


B. AWS Banking Fraud Detection (Nov 2023)

Link: [Banking Fraud Detection with Machine Learning and Real-time Analytics on AWS]

Pipeline & Architecture:
- Leverages Amazon Fraud Detector for account takeover and anti-money laundering solutions.
- Additional services: Amazon Timestream (rule-based), Neptune (graph analytics).
- Alerting via QuickSight dashboards, integrated with OpenSearch for analytics.

Conclusions:
- Combines ready-made ML templates with real-time analytics for quick deployment.
- Enables alerts and analytics dashboards for proactive fraud management.

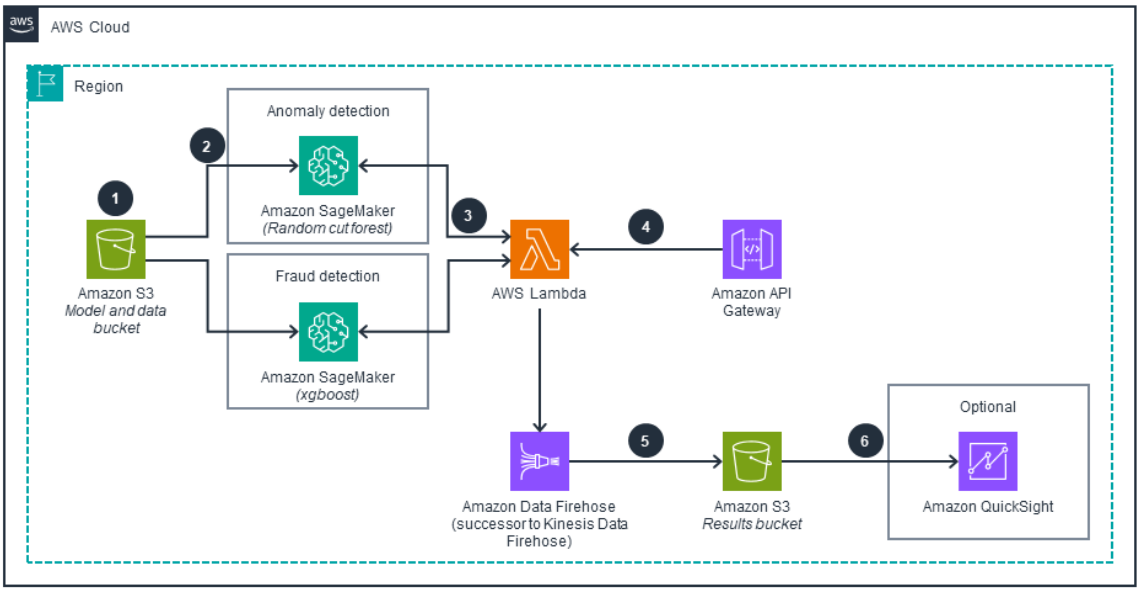


C. Credit Card Real-time Fraud Detection (Guidance by AWS)

Link : [Guidance for Fraud Detection Using Machine Learning on AWS]

Pipeline & Architecture:
- S3-stored transaction datasets.
- Data processing and model training in SageMaker notebooks.
- AWS Lambda invokes SageMaker endpoints for real-time scoring.
- Predictions exposed via API Gateway for ingestion.

Conclusions:
- Provides a robust template for implementing automated, real-time ML pipelines for fraud detection on AWS.



D. "Sandwich" Deep Learning Model — GBDT → GRU → RF

Link : [1711.01434] Transaction Fraud Detection Using GRU-centered Sandwich-structured Model

Pipeline & Architecture:
- Features extracted/optimized through GBDT.
- Sequential features processed via GRU.
- Final classification by Random Forest—"sandwich" setup.
- Applied in banking/insurance contexts, detecting sequential transactional anomalies.

Conclusions:
- Outperformed traditional classifiers and RNN-only models, improving F1 scores by 1.5× and reaching significant accuracy in credit card forgery detection.

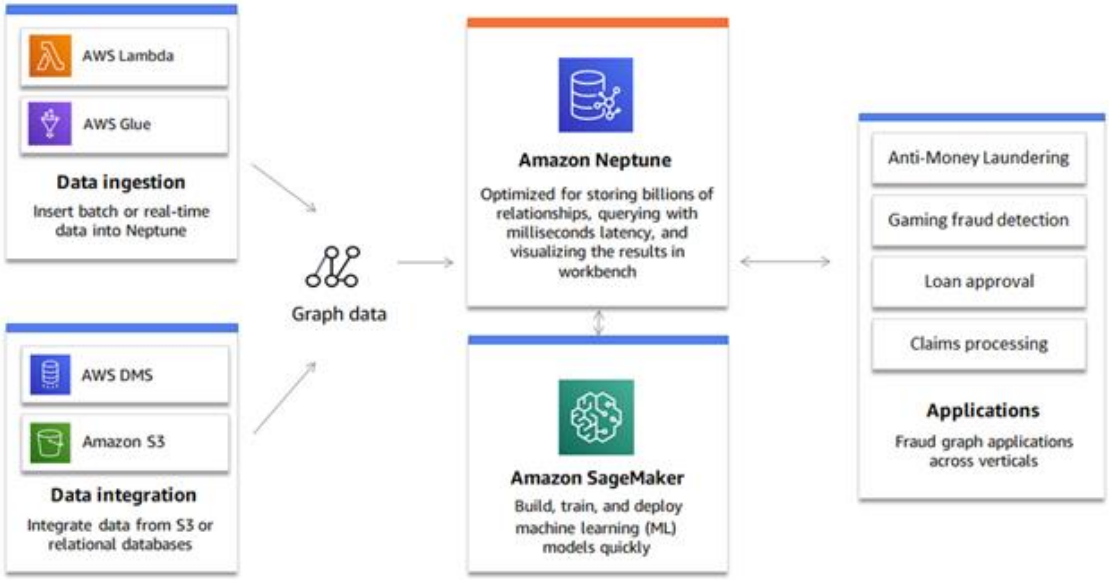
E. Fraud Detection with Knowledge Graphs (Neptune + Bedrock + GNNs)

Link : How AWS Partners can Deliver Anti Fraud Solutions using AWS GenAI and Amazon Neptune Graph Capabilities

Pipeline & Architecture:
- Transaction data → preprocessing → knowledge graph via Amazon Bedrock & LangChain.
- Graph stored in Amazon Neptune; GNN models trained via Neptune ML.
- Fraud inference over the graph structure for predictions.

Conclusions:

- Knowledge graph approach captures complex relationships and network patterns, enabling advanced fraud detection beyond conventional ML.



F. Academic Survey: Real-Time ML Pipelines in E- Commerce and Financial Systems

Link : [Machine Learning Pipeline for Fraud Detection and Prevention in E-Commerce Transactions]

Pipeline & Architecture:
- Real-time transaction monitoring using supervised, anomaly detection, and hybrid ML approaches.
- Emphasis on high-volume, low-latency architectures.
- Highlights ethical, regulatory, and UI/UX considerations.

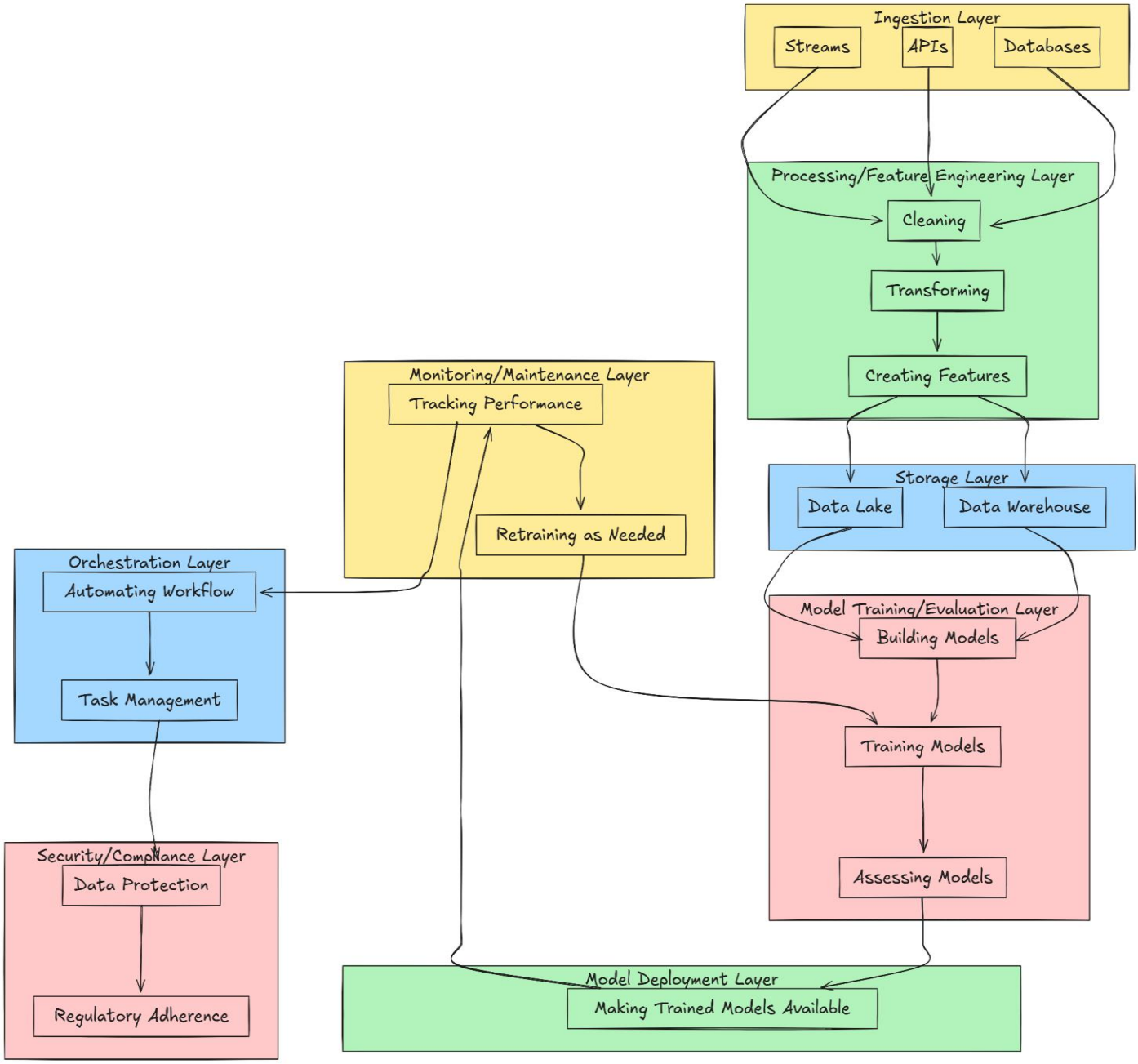
Conclusions:
- AI enables robust fraud protection in real-time systems while demanding attention to fairness, efficiency, and compliance.

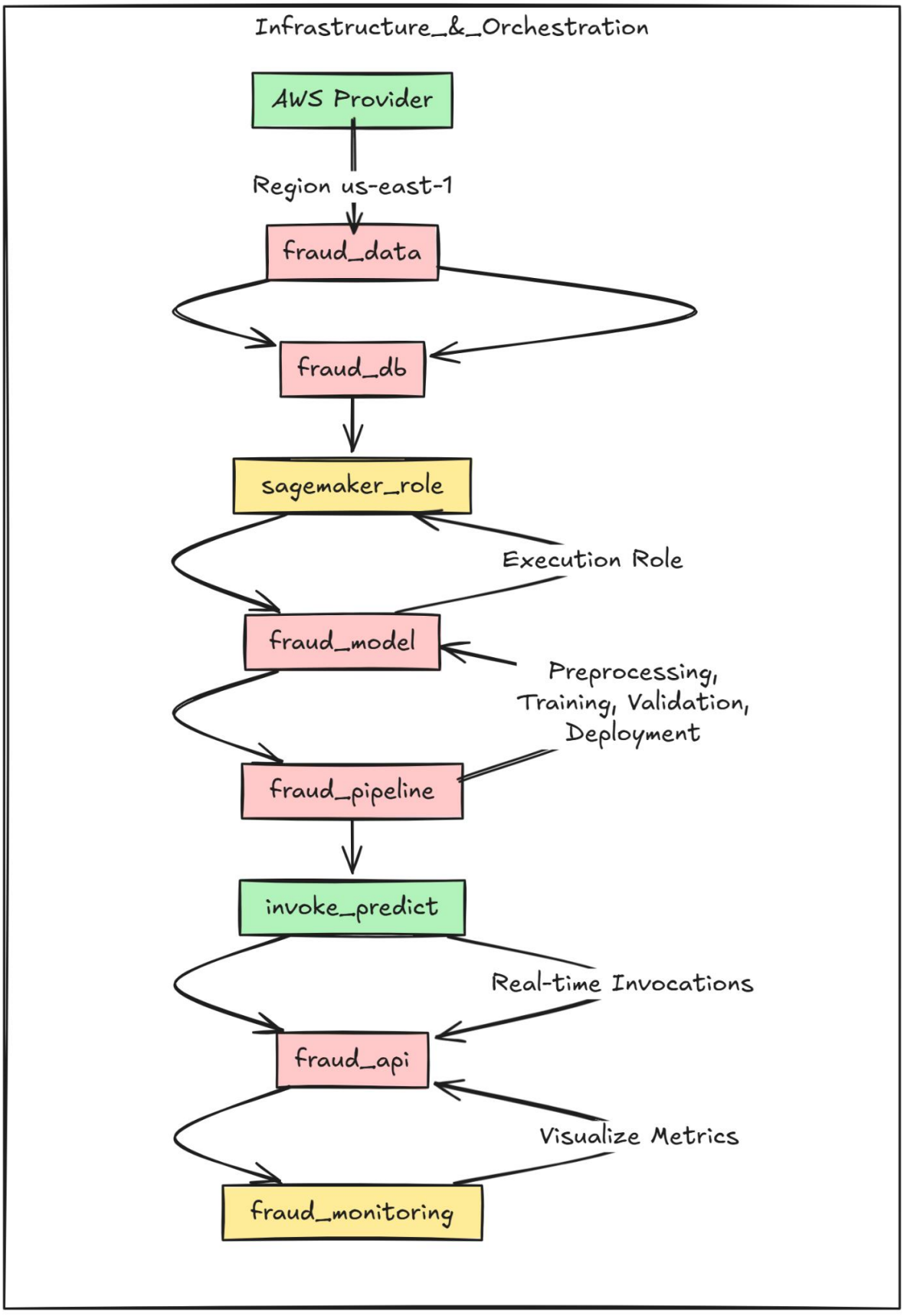
3. Proposed Fraud Detection Pipeline

Here's a design inspired by your current Terraform-AWS setup and the reviewed case studies:

Reference taken from : *Building an End-to-End Fraud Detection Pipeline on AWS: From Data Ingestion to MLOps Automation*

3.1 Infrastructure & Orchestration (Terraform-managed)




Infrastructure_&_Orchestration
AWS Provider
Region us-east-1
fraud_data
fraud_db
sagemaker_role
Execution Role
fraud_model
Preprocessing,
Training, Validation,
Deployment
fraud_pipeline
invoke_predict
Real-time Invocations
fraud_api
Visualize Metrics
fraud_monitoring

3.2 Pipeline Stages & Details

Stage	Components & Functionality
Data Ingestion	Upload raw transaction logs to S3.
ETL & Storage	Use AWS Glue to preprocess, clean, engineer features; load into Redshift or S3 feature store.
Model Training & Selection	Train multiple models (XGBoost, "sandwich" GBDT→GRU→RF, GNN) in SageMaker. Track experiments and tune hyperparameters.
Model Validation	Validate using imbalanced-aware metrics (precision, recall, F1, AUC). Consider sampling techniques like SMOTE.
Deployment	Deploy the best model(s) via SageMaker endpoints; batch inference using API Gateway and Lambda.
Real-Time Inference	Stream or API-driven scoring from new transactions.
Graph Module (Optional)	Build a Neptune-backed graph from transactions. Train GNNs via Neptune ML for network-based detection.
Monitoring	Monitor model performance, drift, and anomalies via CloudWatch; dashboards via QuickSight.
Alerts & Feedback Loop	Trigger alerts via SNS/SQS. Misclassifications fed back for retraining.
Automation	Orchestrate all steps in SageMaker Pipelines and Terraform-managed infrastructure.

4. Conclusions from Each Case Study

Case Study	Key Learning
NSUS Group	SageMaker Pipelines can automate and optimize ML workflows, improving outcomes and reducing manual effort.
AWS Banking Fraud	Prebuilt fraud services (Amazon Fraud Detector) + analytics tools accelerate deployment.
AWS Guidance (Credit Card Flow)	Real-time scoring via Lambda + SageMaker + API Gateway is viable for low-latency systems.
Sandwich Model	Feature stacking and sequential modeling can dramatically enhance detection performance.
Graph-Based (Neptune + GNN)	Knowledge graphs capture relational data, enriching fraud detection beyond independent features.
Academic Real-time ML Survey	Real-time ML is essential—but it must also address fairness, ethics, compliance, and user trust.

5. Final Remarks

Bringing it all together: your Terraform-managed AWS architecture forms a robust foundation. By integrating pipelines inspired by AWS-native real-time strategies, advanced hybrid modeling (e.g., sandwich models), and network insights (graph-based detection), your system can become both scalable and highly effective. The modular design allows experimentation with various approaches while streamlining deployments, monitoring, and feedback — ultimately building a dynamic, ethical, and high-performing fraud detection solution.