|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PROJECT OVERVIEW STATEMENT** | | **Project Name:** AI-Powered Credit Card Fraud Detection: Anomaly-Based Alerting & Case Management System | | **Student Name:** Paul Rohit Jangareddi |
| **Problem/Opportunity:**  Credit card fraud is on the rise, which hurts financial losses and erodes customer trust. The traditional methods fall behind the emerging fraud patterns and are consequently unable to detect fraud or create false alerts. The AI-powered Anomaly Detection System detects suspicious transactions, creates cases, and expedites fraud investigations to improve accuracy, efficiency, and security. | | | | |
| **Goal:**  Design and deploy an AI-enabled credit card fraud detection system within 3 months that reduces fraudulent transactions by at least 30%, ensuring the lowest financial loss and increased fraud investigation efficiency. It would be applied in anomaly detection for flagging suspicious transactions in real time, automation of case creation, and proactive response towards financial institutions.  In this regard, $12.5 billion U.S. consumer online scam losses and $486 billion globally in fraud losses mark the year 2023. It is scalable and AI-driven to increase the accuracy of fraud detection while reducing false positives. Consumer scam losses were up 22% YoY, and therefore, the system will support financial institutions in risk reduction, compliance, and the safety of customers against ever-evolving fraud threats. | | | | |
| **Objectives:**   1. Design and Train the Fraud Detection Model   Output: Design and train an AI-based anomaly detection model to identify suspicious credit card transactions.  Timeframe: End of Month 1  Metric: Detect fraudulent transactions with at least 90% recall, holding the false positive rate below 5%  Action: Employ Isolation Forest, One-Class SVM, and Autoencoders for anomaly detection.   1. Real-Time Fraud Alerting System Implementation   Output: Deploy a system to flag suspicious transactions in real time.  Timeframe: End of Month 2.  Measure: Less than 2-second latency in fraud detection per transaction.  Action: Integrate fraud detection with real-time streaming pipeline utilising Apache Kafka or AWS Lambda.   1. Automation of Case Creation and Investigation Workflow   Outcome: To provide a case management system where the flagged transactions are logged for investigation automatically.  Time Frame: Mid of Month 2.  Measure: 100% of the flagged transactions log cases with relevant details to be reviewed.  Action: Automate the case database and integrate it with an investigation dashboard for fraud analysts.   1. Optimize Fraud Detection System for Accuracy and Efficiency   Pursue the testing and optimization of the fraud detection system based on real-time data.  Time Frame: End of Month 2.  Measure: Reduce fraudulent transaction losses by 30% compared to historical baseline.  Action: Conduct pilot testing and fine-tune the model using real or simulated transaction data.   1. Deploy and Monitor the Fraud Detection System   Outcome: fraud detection system will be deployed into a production-ready state, with monitoring, to ensure performance.  Time Frame: End of Month 3.  Measure: Continuous fraud detection performance; retrain weekly for new fraud trends.  Action: Deploy on cloud infrastructure-AWS/GCP/Azure-with real-time monitoring tools implemented to track the performance and trends. | | | | |
| **Success Criteria:**   * Accuracy: The model should accurately detect at least 90% of fraudulent transactions with a 5% false positive rate. * Automation: 100% of the flagged transactions should automatically create a case for investigation. * Customer Impact: Reduce fraudulent transaction losses by at least 30% from baseline data, with increased security and customer trust. * Proficiently Demonstrated Skills: machine learning, processing of data in real time, development of the case management system. | | | | |
| **Assumptions, Risks and Obstacles:**  **Assumptions:**   * Data Availability: Sufficient and high-quality historical transaction data will be provided for model training and testing. * Stakeholder Engagement: Timely feedback and collaboration with fraud analysts and financial institutions for system validation and optimization. * Infrastructure: Access to necessary cloud infrastructure (AWS/GCP/Azure) for deployment and real-time monitoring.   **Risks:**   * Data Quality Issues: Incomplete, unclean, or biased data may affect model performance, leading to inaccurate fraud detection results.   Mitigation: Ensure thorough data preprocessing, normalization, and validation before model training.   * Model Accuracy: The fraud detection model may fail to achieve the desired 90% recall or result in high false positives.   Mitigation: Perform rigorous model evaluation and iterative tuning to optimize accuracy and reduce false positives.   * Integration Challenges: Difficulty in integrating the fraud detection system with existing transaction processing pipelines or case management systems.   Mitigation: Allocate time for testing and troubleshooting integration, with clear documentation and collaboration with technical teams.   * Real-Time Processing Delays: Latency issues could arise when processing large volumes of transactions, affecting the real-time nature of fraud detection.   Mitigation: Optimize the system architecture for scalability, using efficient streaming technologies like Apache Kafka or AWS Lambda.   * Regulatory Compliance: Ensuring the fraud detection system complies with financial regulations (e.g., GDPR, PCI-DSS) may pose challenges.   Mitigation: Involve compliance experts during the development and deployment phases to ensure the system adheres to legal standards.  **Obstacles:**   * Limited Training Data for Anomalies: The scarcity of fraudulent transaction data could make it difficult to train an accurate model.   Mitigation: Use synthetic data generation or transfer learning techniques to augment the training dataset.   * Resource Constraints: Limited access to technical resources (e.g., cloud services, computing power) or personnel may cause delays.   Mitigation: Plan ahead for resource allocation, ensure availability of necessary tools, and prioritize key tasks.   * Changing Fraud Patterns: Evolving fraud tactics might outpace the model’s ability to detect new schemes.   Mitigation: Regularly update the model through retraining and monitoring based on new fraud patterns. | | | | |
| **Prepared By:**  Paul Rohit Jangareddi | **Date:**  01/27/2025 | | **Approved By** | **Date** |